Fast campus

2017 Machine Learning with R

Clustering

강필성

고려대학교 산업경영공학부

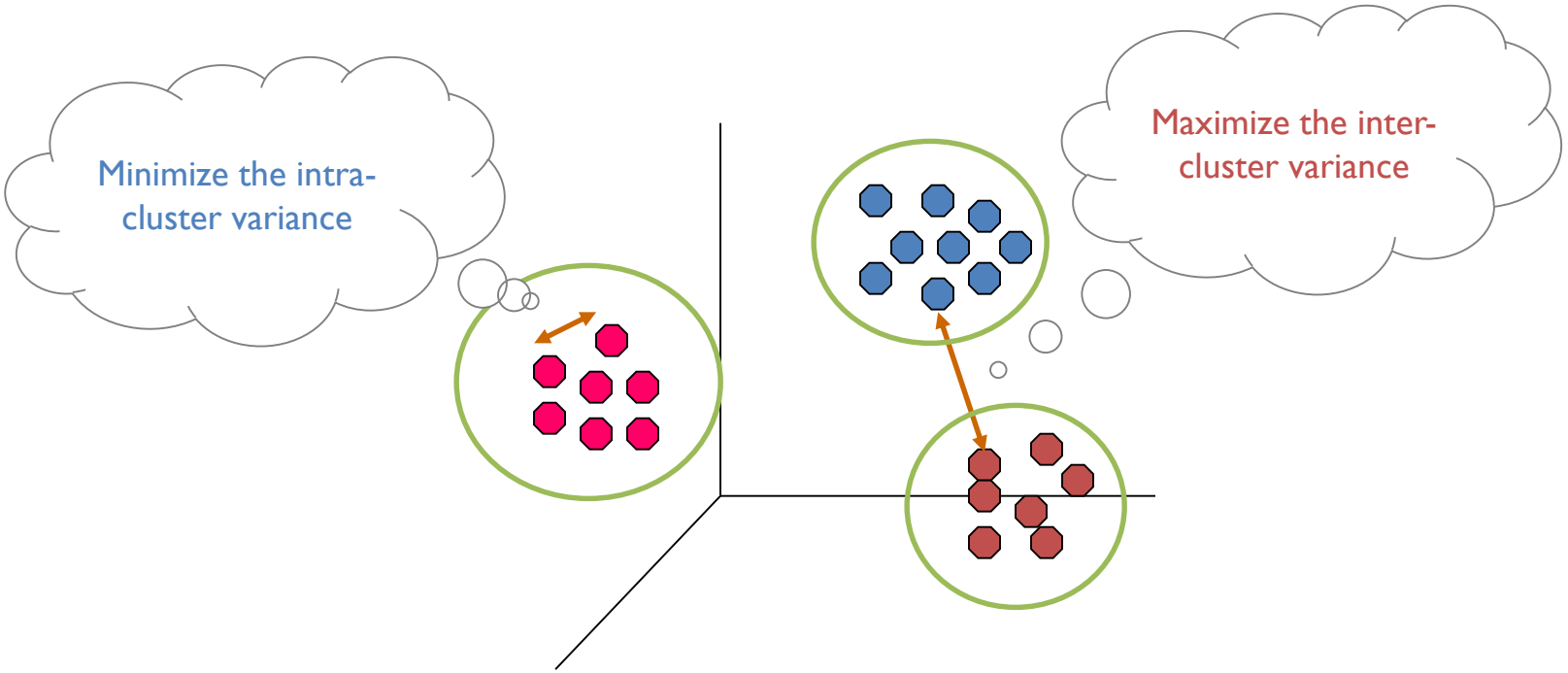
pilsung_kang@korea.ac.kr

목차

군집화: Clustering

❖ 군집화(Clustering)

- 관측치들의 집단을 판별
  - ✓동일한 집단에 소속된 관측치들은 서로 유사할수록 좋음
  - ✓상이한 집단에 소속된 관측치들은 서로 다를수록 좋음

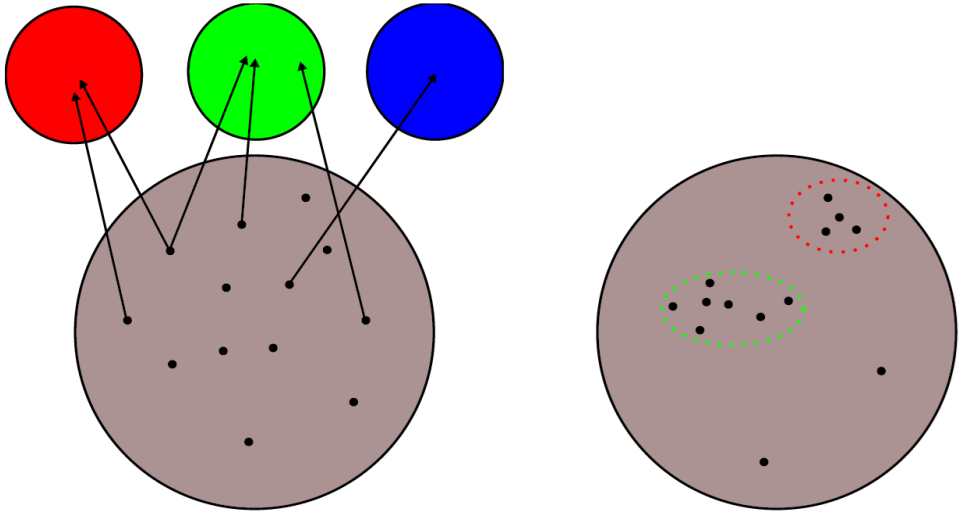


군집화: Clustering

Andrews and Fox (2007)

❖ 분류 (Classification) vs. 군집화(Clustering)

- 분류(Classification): 범주의 수 및 각 개체의 범주 정보를 사전에 알 수 있으며, 개체의 입력 변수 값들로부터 범주 정보를 유추하여 새로운 개체에 대해 가장 적합한 범주로 할당하는 문제 (supervised learning)
- 군집화(Clustering): 군집의 수, 속성, 멤버십 등이 사전에 알려져 있지 않으며 최적의 구분을 찾아가는 문제 (unsupervised learning)



(a) Classification (b) Clustering

군집화: 적용사례

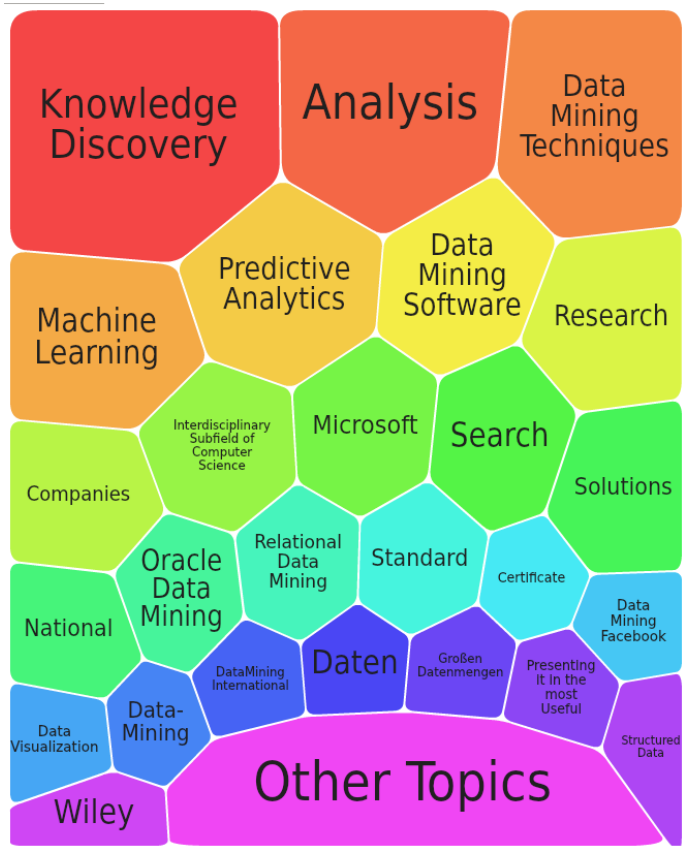
❖ 군집화 적용 사례

▪ 데이터에 대한 이해

- ✓ 웹브라우징 시 유사한 문서들을 표시
- ✓ 유사한 기능을 수행하는 유전자/단백질 집합
- ✓ 유사한 추세를 나타내는 주식 종목들 등

Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%) , organizations (43%) Ahavat Israel - The Amazing Jewish Website! Israel and Judaism Judaica Collection
2 View Results Refine Query Based On This Cluster	15	Ministry of Foreign Affairs (33%), Ministry (87%) Publications and Data of the BANK OF ISRAEL Consulate General of Israel to the Mid-Atlantic Region The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%) Interactive Israel tourism guide - Jerusalem Ambassade d'Israel Travel to Israel Opportunites
4 View Results Refine Query Based On This Cluster	7	Middle East (57%), History (57%); WAR (42%) , Region (42%) , Complete (42%) , Listing (42%) , country (42%) Israel at Fifty: Our Introduction to The Six Day War Machal - Volunteers in the Israel's War of Independence HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	Economy (68%), Companies (55%), Travel (55%) Israel Hotel Association Israel Association of Electronics Industries Focus Capital Group - Israel

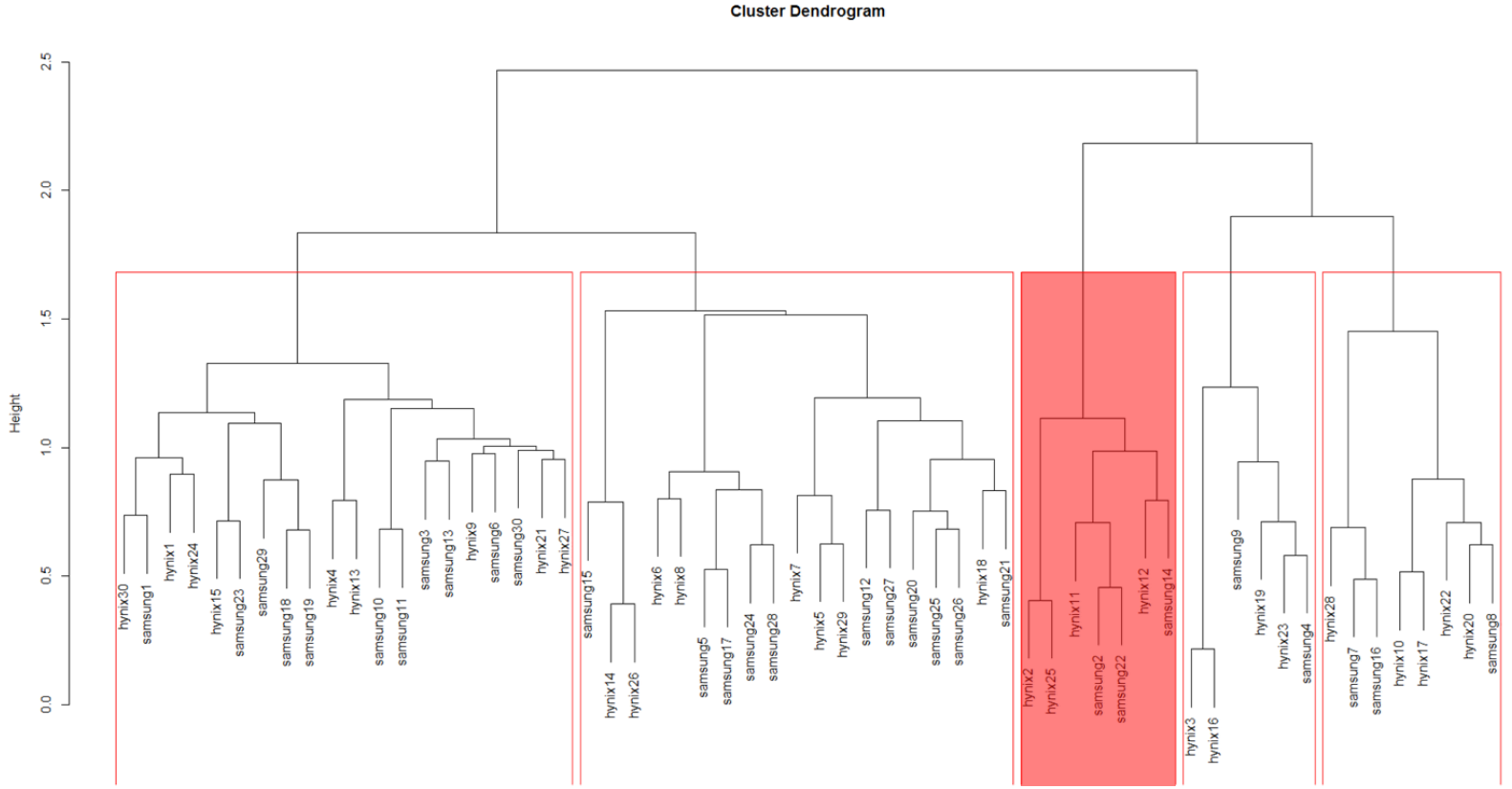


군집화: 적용사례

❖ 군집화 적용 사례

▪ 전략 수립

✓ 경쟁사와의 특허 문서 분석을 통한 장단점 파악



군집화: 적용사례

❖ 군집화 적용 사례

▪ 전략 수립

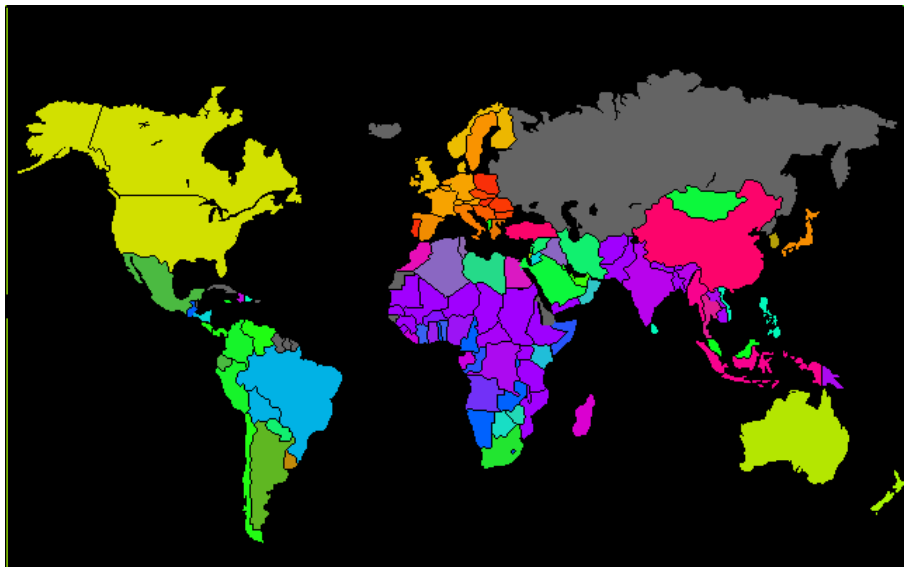
✓ 경쟁사와의 특허 문서 분석을 통한 장단점 파악

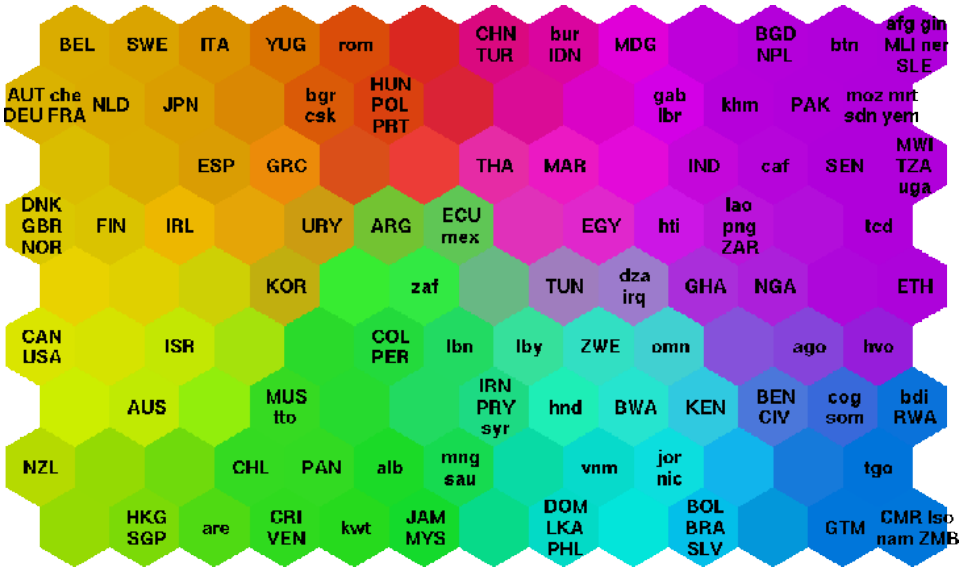
	회사	일련번호	특허명	초록
2	SK하이닉스	2	멀티 레귤레이터 회로 및 이를 구비한 집적회로	본 기술에 따른 레귤레이터 회로는, 입력전압을 일정한 전압 레벨로 레귤레이팅하여 출력하도록 구성된 레귤레이터 및 복수개의 전압 생성 코드 들에 의해 결정되는 내부 저항값들에 따라 상기 레귤레이터의 출력 전압을 분배한 분배전압들을 각각 출력하도록 구성된 복수개의 전압 분배회로를 포함한다.
3	SK하이닉스	11	내부 전압 생성 회로 및 그의 동작 방법	펌핑 동작을 통해 내부 전압을 생성하는 내부 전압 생성 회로에 관한 것으로, 다수의 펌핑부를 포함하며, 목표 전압 레벨에 대응하는 최종 펌핑 전압을 생성하기 위한 펌핑 전압 생성부, 및 상기 목표 전압 레벨에 대응하여 상기 다수의 펌핑부의 활성화 개수를 제어하기 위한 활성화 제어부를 구비하는 내부 전압 생성 회로가 제공된다.
4	SK하이닉스	12	자기 메모리 장치를 위한 라이트 드라이버 회로 및 자기 메모리 장치	비트라인과 소스라인 간에 접속되며, 비트라인 방향으로 인접하는 한 쌍의 자기 메모리 셀이 소스라인을 공유하는 복수의 자기 메모리 셀로 이루어진 메모리 셀 어레이를 포함하는 자기 메모리 장치를 위한 라이트 드라이버 회로로서, 정의 기록전압 공급단자와 부의 기록전압 공급단자 간에 접속되어, 라이트 인에이블 신호 및 데이터 신호에 따라 정의 기록전압 또는 부의 기록전압에 의한 전류를 비트라인에 선택적으로 공급하는 스위칭부를 포함하는 자기 메모리 장치를 제공한다.
5	SK하이닉스	25	전압 레귤레이터 및 전압 레귤레이팅 방법	전압 레귤레이터는 출력전압을 전압 출력단으로 출력하는 전압 출력부와, 제1 제어코드의 제어에 따라 분배 저항값을 조절하는 제1 저항분배 스테이지와, 제1 저항분배 스테이지에서 결정된 분배 저항값을 제2 제어코드의 제어에 따라 조절하는 제2 저항분배 스테이지;를 포함하며, 전압 출력단을 통해서 출력되는 출력전압의 전압레벨은 제1 및 제2 저항분배 스테이지를 통해서 결정된 상기 분배 저항값과, 기준저항의 저항값 비율에 따라 조절되는 것을 특징으로 한다.
6	삼성전자	2	전압 공급 장치 및 그것을 포함한 불휘발성 메모리 장치	본 발명에 따른 전압 공급 장치는 전원 전압을 승압하고, 상기 승압된 전압을 출력 라인으로 제공하기 위한 전하 펌프 및 상기 출력 라인의 전압 레벨을 목표 전압 레벨로 유지하기 위한 전압 제어 회로를 포함한다. 본 발명에 따른 상기 전압 제어 회로는 웰 상에 형성된 제 1 영역 및 제 2 영역을 포함하고, 상기 제 1 영역 및 제 2 영역 사이의 리치 스루(reach through)를 이용하여 상기 출력 라인의 전압 레벨을 제어하기 위한 리치 스루 소자를 포함한다.
7	삼성전자	14	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치가 개시된다. 본 발명의 제 1 실시예에 따른 반도체 메모리 장치는 파워 공급 회로, 스위치들 및 선택기들을 구비한다. 파워 공급 회로는 상기 블록들의 메모리 셀들에 사용되는 제 1 전압 및 제 2 전압을 생성한다. 스위치들은 상기 파워 공급 회로와 상기 제 1 전압이 전달되는 제 1 라인 및 상기 제 2 전압이 전달되는 제 2 라인으로 연결되고, 제어 신호에 응답하여 상기 제 1 전압 및 제 2 전압 중 하나를 대응되는 블록으로 인가한다. 선택기들은 블록 선택 신호 및 디스차아지 성공 신호에 응답하여, 상기 제어 신호를 생성한다. 본 발명에 따른 파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치는 셀 블록마다 별도의 파워 스위치를 구비함으로써 파워 공급 회로의 동작 시간 및 동작 전류를 감소시킬 수 있다. 또한, 기입 전압을 디스차아지한 후 다른 레벨의 전압을 공급함으로써, 상 변화 메모리 장치의 오작동이 방지될 수 있다.
8	삼성전자	22	전압 안정화 장치 및 그것을 포함하는 반도체 장치 및 전압 생성 방법	본 발명은 전압 안정화 장치 및 그것을 이용하는 반도체 장치에 관한 것이다. 본 발명의 기술적 사상의 실시예에 따른 전압 안정화 장치는 제 1 전압을 생성하는 제 1 레귤레이터 및 상기 제 1 전압보다 낮은 제 2 전압을 생성하는 제 2 레귤레이터를 포함하되, 상기 제 2 레귤레이터는 상기 제 1 전압의 레벨과 미리 정해진 기준 전압의 레벨의 비교 결과에 기초하여 상기 제 1 전압 또는 상기 제 1 전압보다 높은 제 3 전압을 선택적으로 이용하여 상기 제 2 전압을 생성한다. 본 발명의 기술적 사상의 실시 예에 따르면 제 1의 전압≥제 2의 전압의 관계를 유지하면서, 동시에 제 2의 전압을 고속으로 전위 변환 시킬 수 있다.

군집화: 적용사례

❖ 군집화 적용 사례

- 대량의 데이터에 대한 요약
  - ✓ 고차원의 데이터를 저차원으로 축약하여 정보를 요약
- 시각화(Visualization)과 밀접한 관계가 있음

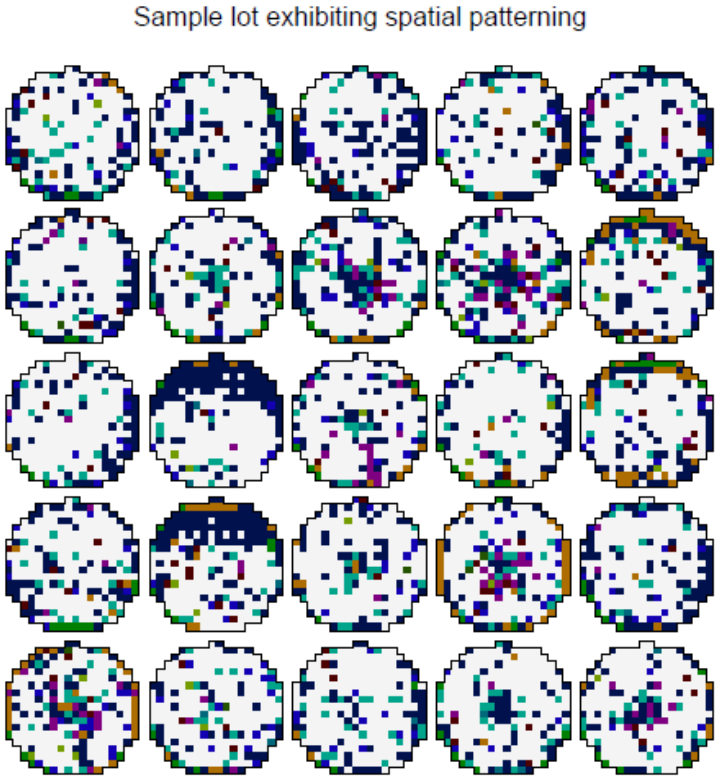


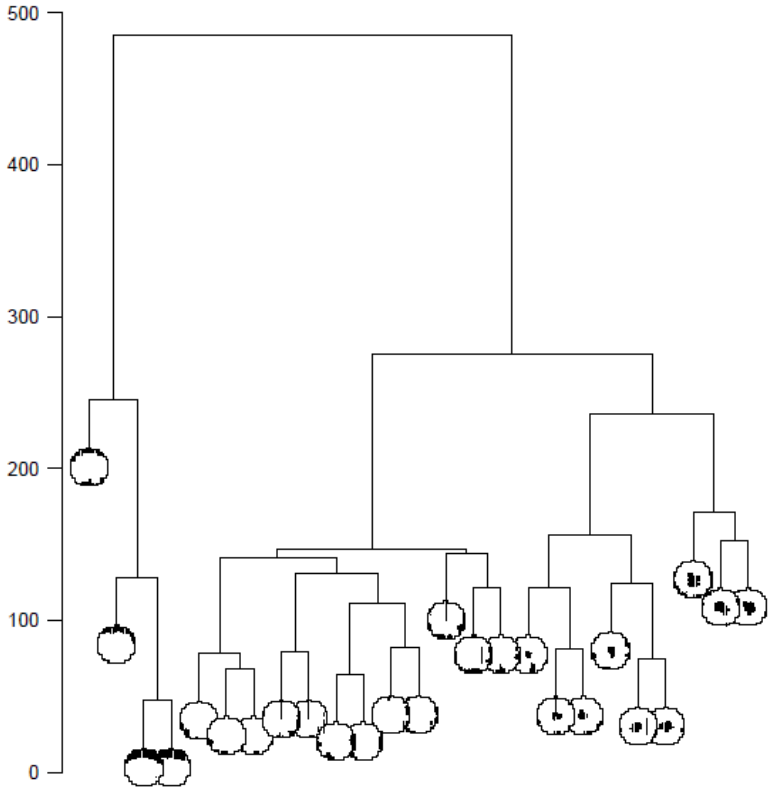


군집화: 적용사례

❖ 군집화 적용 사례

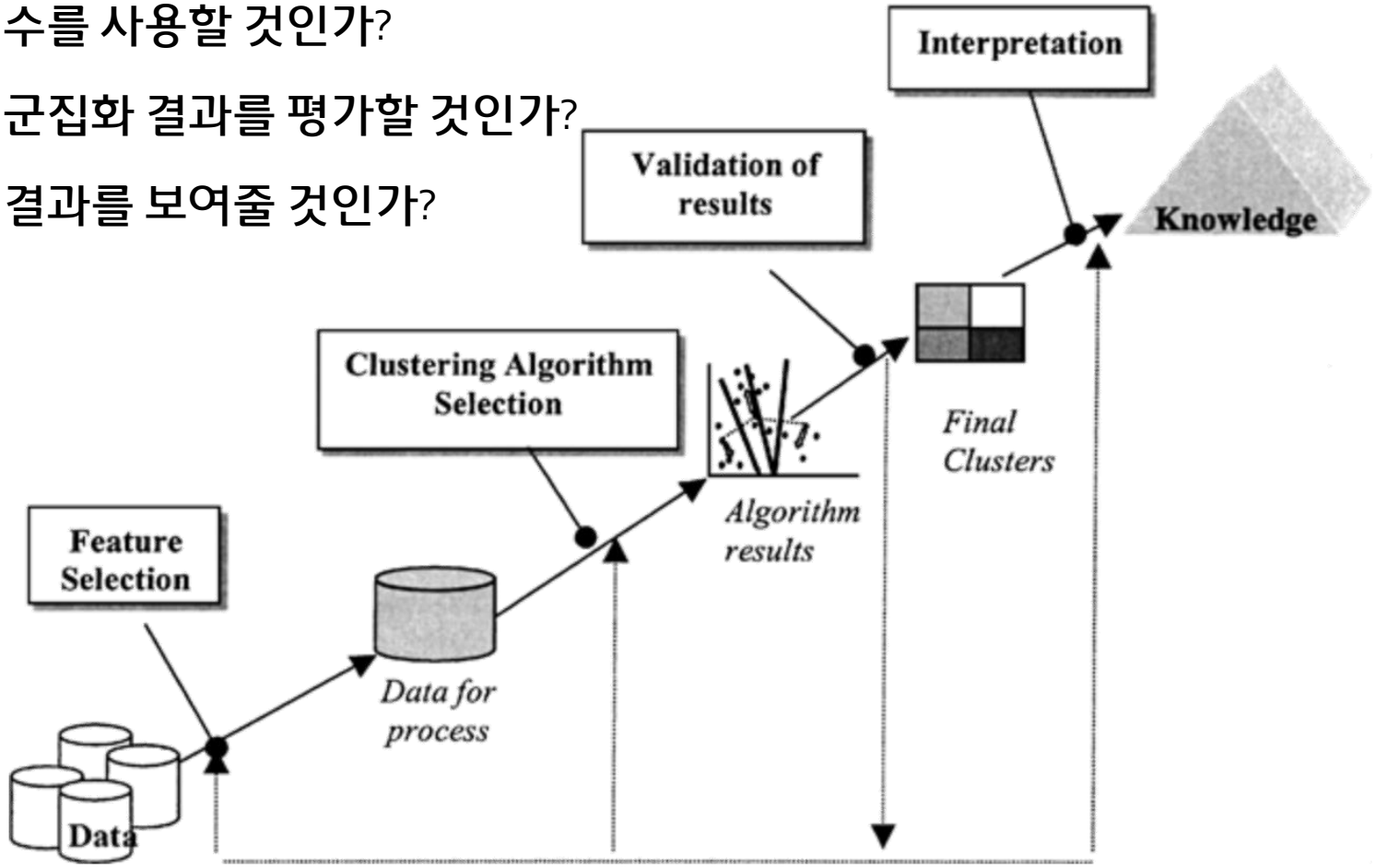
- 웨이퍼 Fail bit map 군집화





군집화: 수행 절차

- 일반적인 군집화 수행 절차
  - 어떤 변수를 사용할 것인가?
  - 어떻게 군집화 결과를 평가할 것인가?
  - 어떻게 결과를 보여줄 것인가?



http://web.itu.edu.tr/sgunduz/courses/verimaden/paper/validity_survey.pdf

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 최적의 군집 수 결정

How many clusters?

Six Clusters

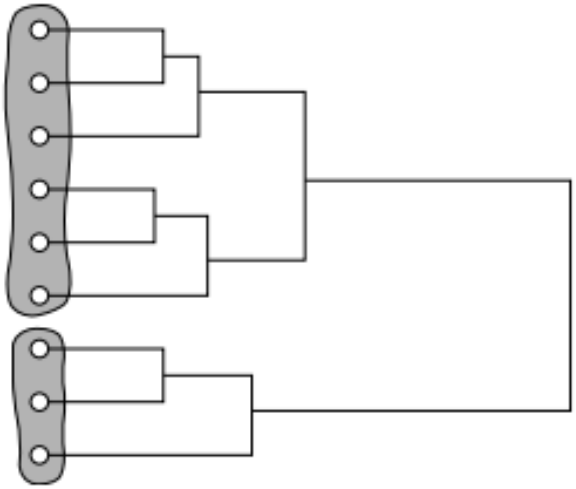
Two Clusters

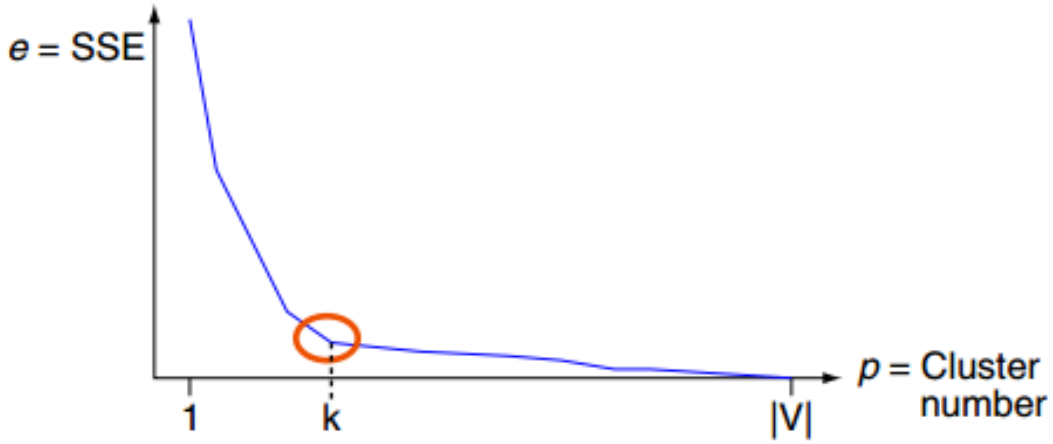
Four Clusters

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 최적의 군집 수 결정
  - ✓ 다양한 군집 수에 대해 성능 평가 지표를 도시하여 최적의 군집 수 선택
  - ✓ Elbow point에서 최적 군집 수가 결정되는 경우가 일반적임





군집화: 고려 사항

- 군집화 수행 시 고려 사항
  - 군집화 결과를 어떻게 평가할 것인가?
  - 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure 부재
- 군집화 평가 지표는 다음과 같이 세 가지 카테고리로 구분할 수 있음
  - External: 정답 레이블과의 비교를 통해 성능 평가 (현실적으로 불가능)
  - Internal: "군집이 얼마나 컴팩트한가"에 보다 초점을 둠
  - Relative: "군집이 얼마나 컴팩트한가"와 "군집끼리 얼마나 다른가"를 동시에 고려하고자 함

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

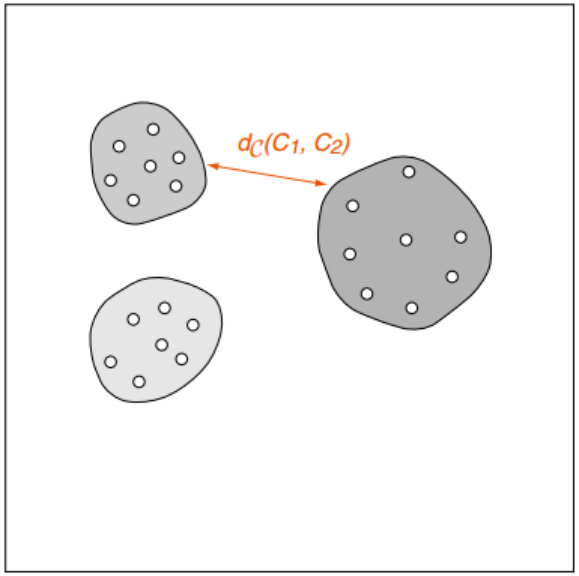
- 군집화 결과를 어떻게 평가할 것인가?
- 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure 부재

External	Internal	Relative
Rand Statistic	Cophenetic Correlation Coefficient	Dunn family of indices
Jaccard Coefficient	Sum of Squared error (SSE)	Davies-Bouldin (DB) index
Folks and Mallows index	Cohesion and separation	Semi-partial R-squared
(Normalized) Hurbert Γ statistic		SD validity index
		Silhouette

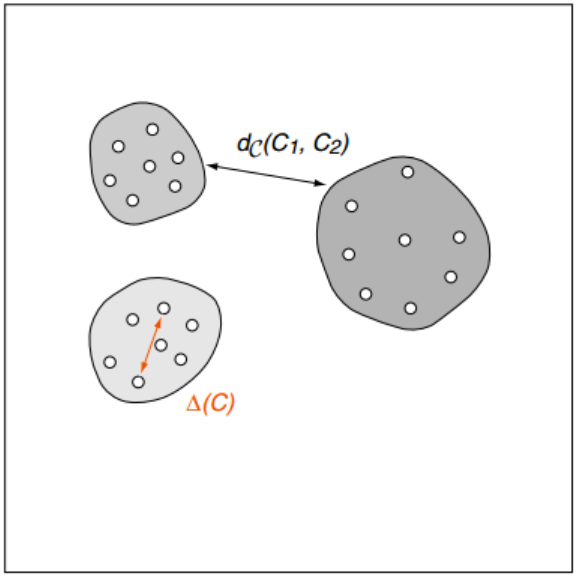
군집화: 고려 사항

- 군집화 평가를 위해 필요한 세 가지 지표 정의
  - 군집화가 잘 되어 있다면
    - ✓(1)번 지표의 값은 크고 (2)번과 (3)번 지표의 값은 상대적으로 작게 나타날 것임

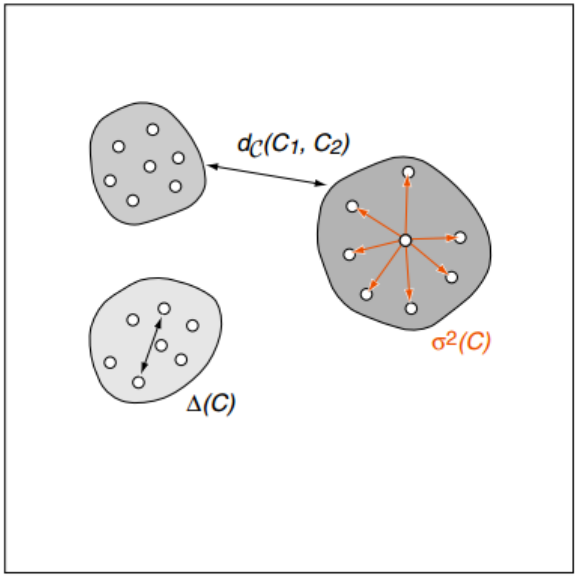
(1) Distance between two clusters



(2) Diameter of a cluster



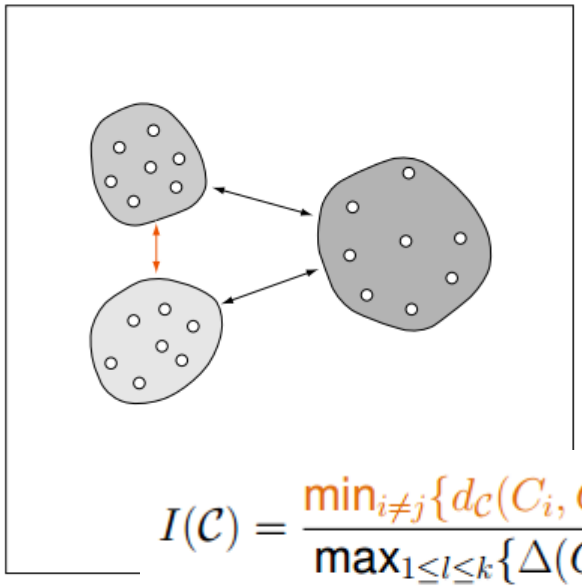
(3) Scatter within a cluster (SSE)



군집화: 고려 사항

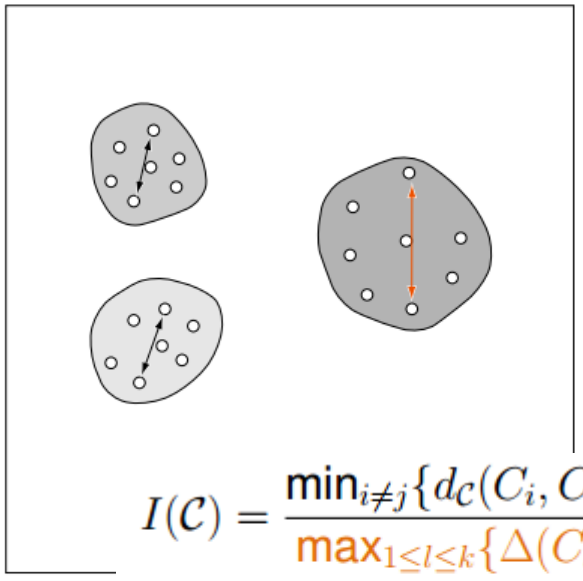
❖ 군집화 평가 지표 1: Dunn Index

- Dunn Index는 군집 내 거리((1)번 지표)중 가장 작은 값을 분자로, 군집의 지름((2)번 지표) 중 가장 큰 값을 분모로 정의함
- Dunn Index는 클수록 우수한 군집화 결과라고 할 수 있음



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

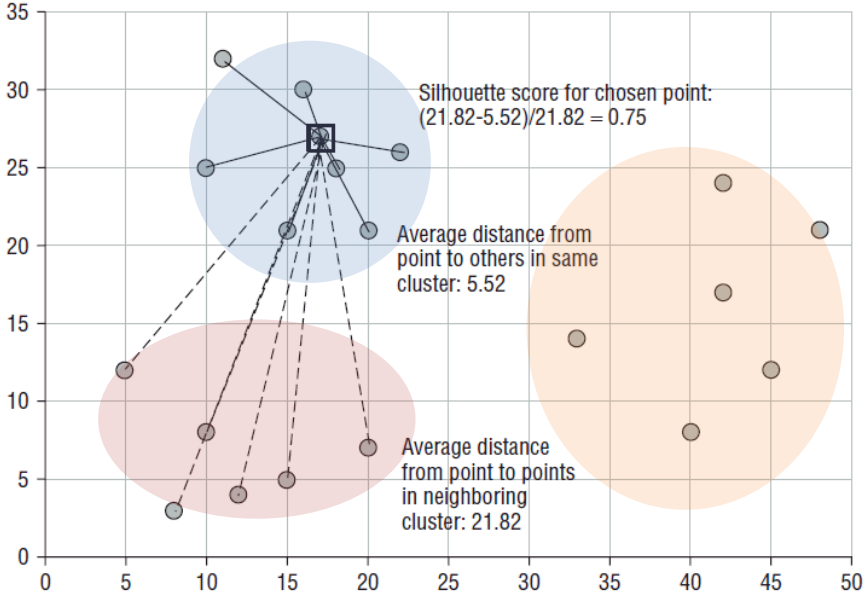
군집화: 고려 사항

❖ 군집화 평가 지표 2: Silhouette

- a(i): 개체 i로부터 같은 군집 내에 있는 모든 다른 개체들 사이의 평균 거리
- b(i): 개체 i로부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 가장 작은 값

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & if\ a(i) < b(i) \\ 0, & if\ a(i) = b(i) \\ b(i)/a(i) - 1, & if\ a(i) > b(i) \end{cases}$$



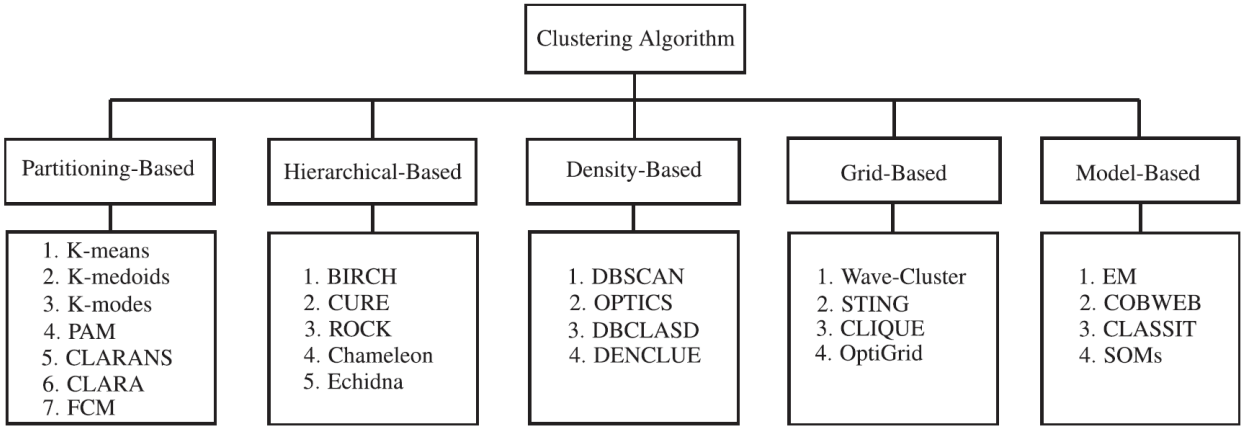
군집화 종류

Fahad et al. (2014)

❖ Hard Clustering vs. Soft Clustering

▪ Hard Clustering (Crisp Clustering)

- ✓서로 겹치지 않는(non-overlapping) 군집 생성
- ✓각 개체는 오직 하나의 군집으로만 할당됨



▪ Soft Clustering (Fuzzy Clustering)

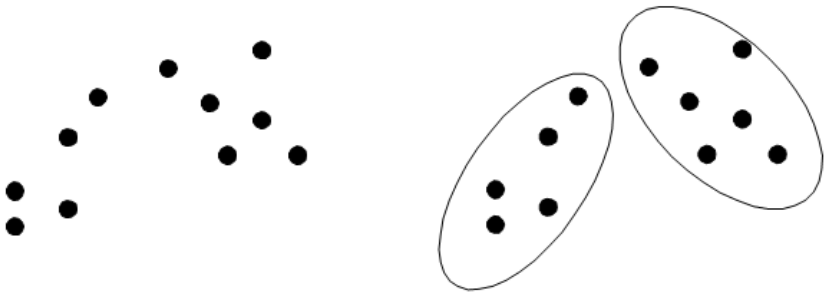
- ✓겹치는 군집을 생성하는 것도 가능함
- ✓한 개체는 여러 개의 군집에 확률적인 할당이 될 수 있음

군집화: 알고리즘

❖ 군집화 알고리즘의 종류

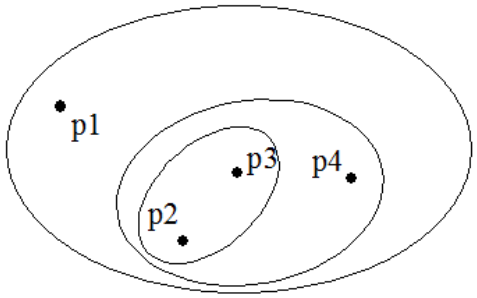
▪ 분리형 군집화

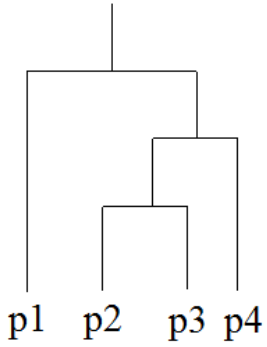
- ✓ 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- ✓ 각 개체들은 사전에 정의된 군집 수 중 하나에 속하는 결과를 도출함



▪ 계층적 군집화

- ✓ 개체들을 가까운 집단부터 차근차근 묶어가는 방식
- ✓ 군집화 결과 뿐만 아니라 유사한 개체들이 결합되는 절차(dendrogram)도 생성

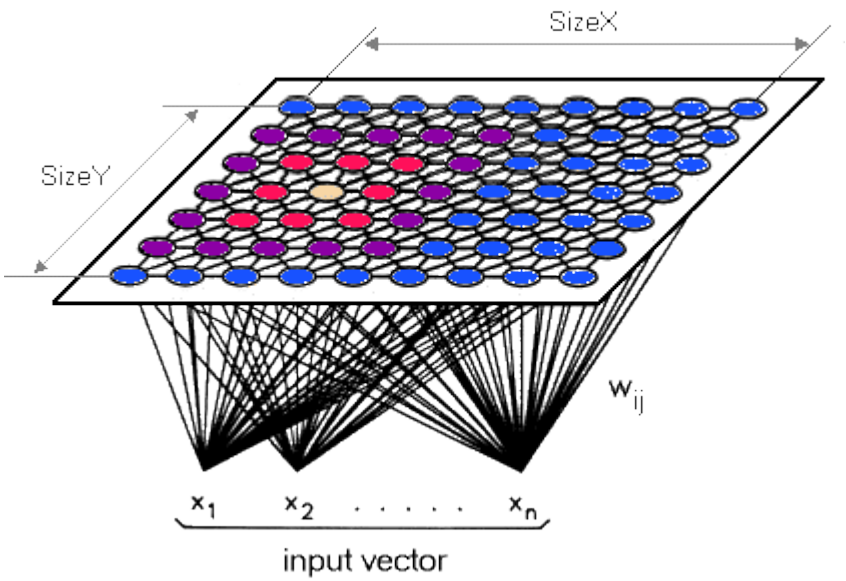




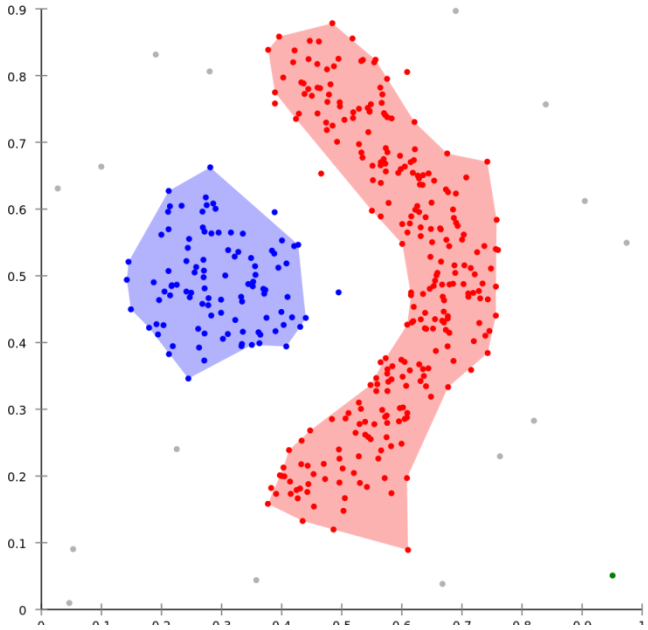
군집화: 알고리즘

❖ 군집화 알고리즘의 종류

- 자기조직화 지도: Self-Organizing Map (SOM)
  - ✓2차원의 격자에 각 개체들이 대응하도록 인공신경망과 유사한 학습을 통해 군집 도출



- 밀도 기반 군집화
  - ✓ 데이터의 분포를 기반으로 높은 밀도를 갖는 세부 영역들로 전체 영역을 구분



목차

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화

- 대표적인 분리형 군집화 알고리즘
  - ✓ 각 군집은 하나의 중심(centroid)을 가짐
  - ✓ 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 생성
  - ✓ 사전에 군집의 수 K가 정해저야 알고리즘을 실행할 수 있음

$$\mathbf{X} = C_1 \cup C_2 \ldots \cup C_K, \quad C_i \cap C_j = \phi, \quad i \neq j$$

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathbf{c}_i||^2$$

K-평균 군집화: K-Means Clustering

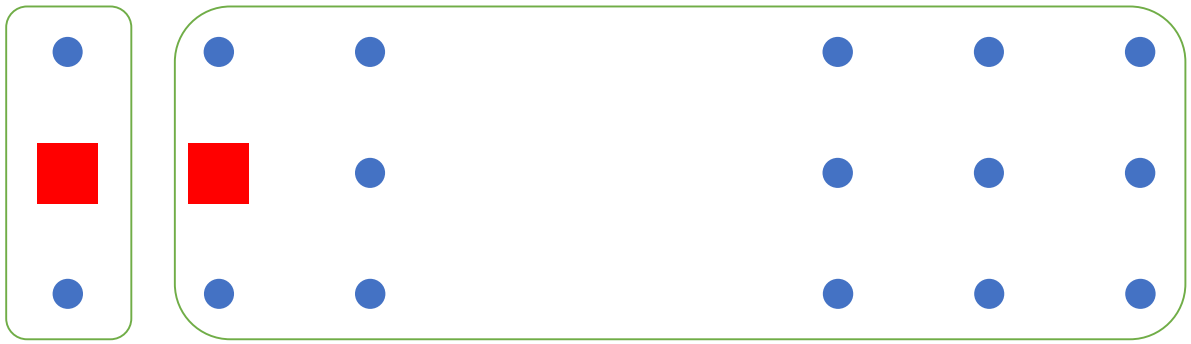
❖ K-평균 군집화 (K-Means Clustering) 수행 절차

- 1단계: K개의 초기 군집 중심(initial centroid) 설정
- 2단계: 다음 절차를 반복
  - ✓모든 개체를 가장 가까운 군집 중심에 할당하여 군집 구성
  - ✓할당된 개체들을 이용하여 군집 중심을 재설정
  - ✓종료 조건: 모든 군집 중심의 위치가 변하지 않고, 모든 개체의 군집 할당 결과에 변화가 없을 때 알고리즘 종료
- Note: 초기 중심은 종종 무작위로 선택되며 따라서 군집화의 결과가 초기 중심 설정에 따라 다르게 나타나는 경우가 발생할 수도 있음

K-평균 군집화: K-Means Clustering

- K-평균 군집화 (K-Means Clustering) 수행 절차 예시
  - 1단계: K개의 초기 군집 중심(initial centroid) 설정

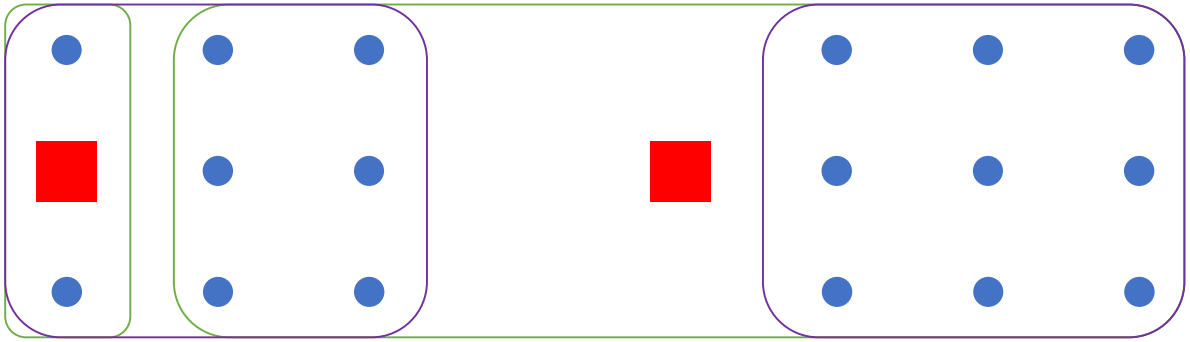
  - 2-1단계(1회차): 모든 개체를 가장 가까운 중심에 할당
  - 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정



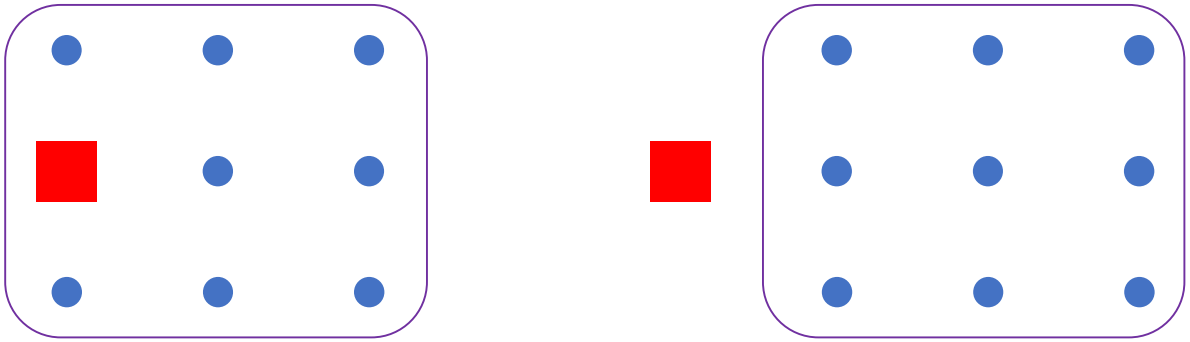
K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차 예시

- 2-1단계(2회차): 모든 개체를 가장 가까운 중심에 할당



- 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정

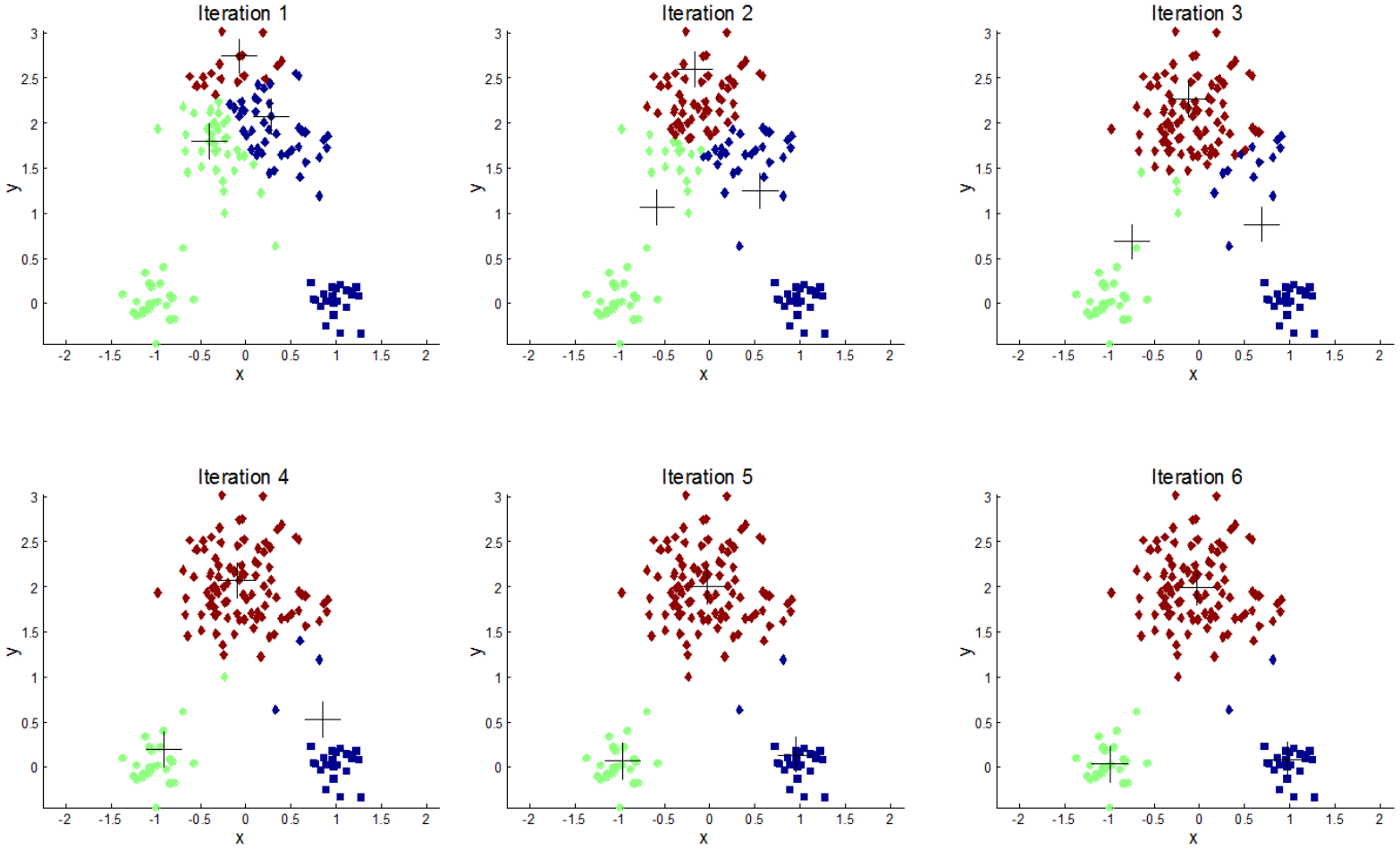


- 군집 중심과 개체 할당에 변화가 없으므로 알고리즘 종료

K-평균 군집화: K-Means Clustering

❖ 초기 중심 설정의 영향

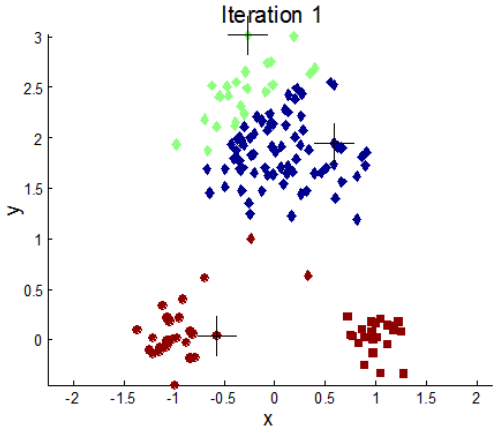
- 바람직한 결과

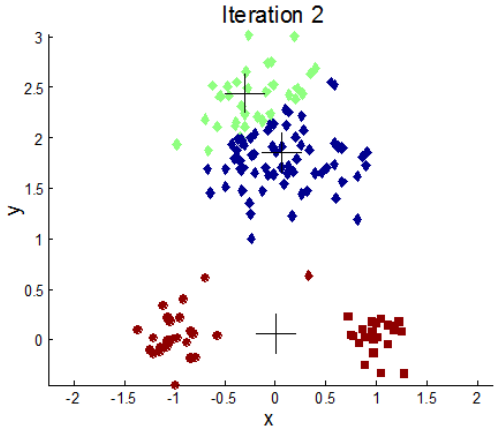


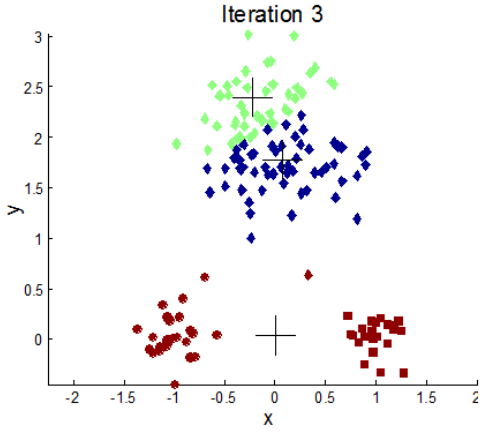
K-평균 군집화: K-Means Clustering

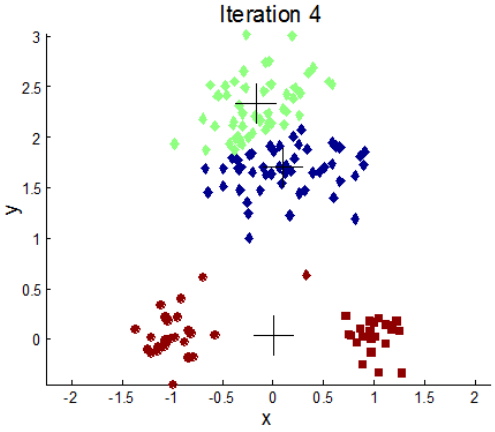
❖ 초기 중심 설정의 영향

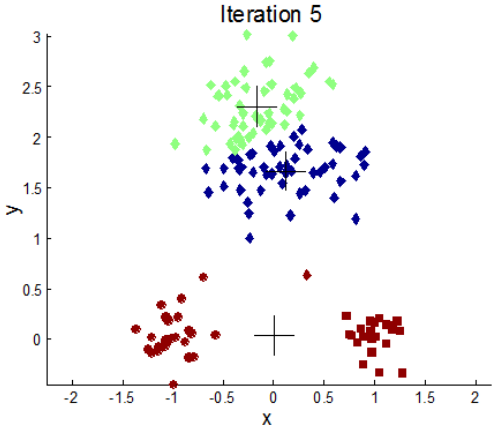
- 바람직하지 않은 결과





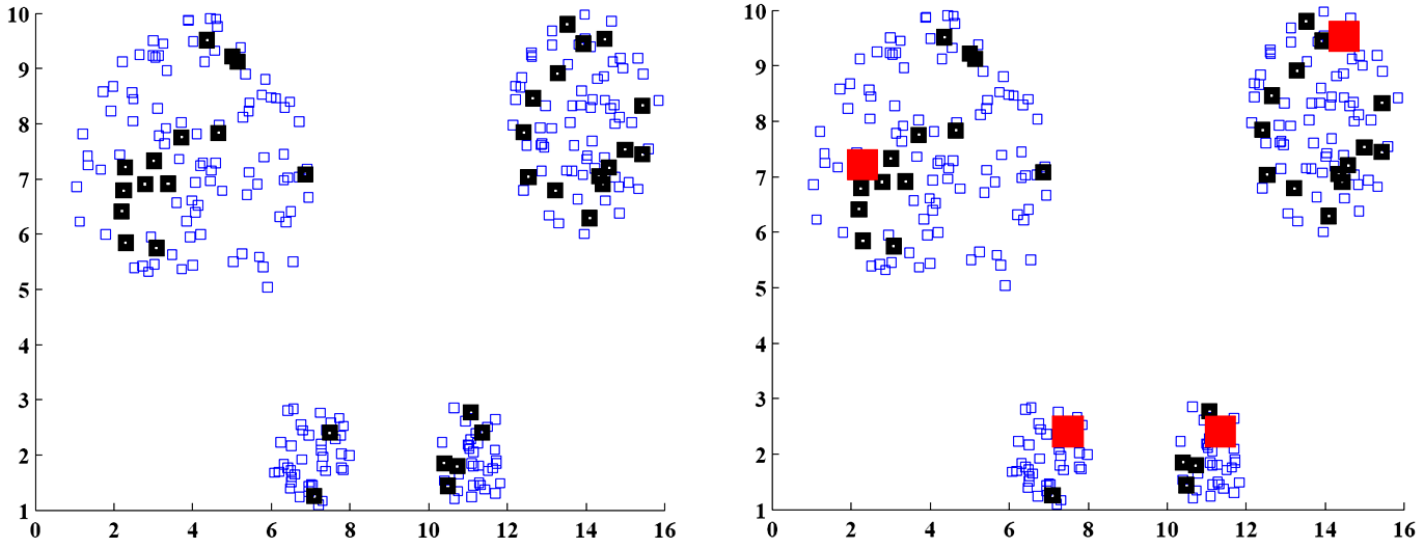






K-평균 군집화: K-Means Clustering

- 무작위 초기 중심 설정의 위험을 감소하고자 하는 다양한 연구가 존재
  - 반복적으로 수행하여 가장 여러 번 나타나는 군집을 이용
  - 전체 데이터 중 일부만 샘플링하여 계층적 군집화를 수행한 뒤 초기 군집 중심 설정
  - 데이터 분포 $\mathcal{L}(\mathbf{x}_s|\mathbf{S},\mathbf{C}) = d_G(\mathbf{x}_s,\mathbf{S}) \times \frac{1}{1+\exp\left(-d_R(\mathbf{x}_s,\mathbf{S})\right)}$

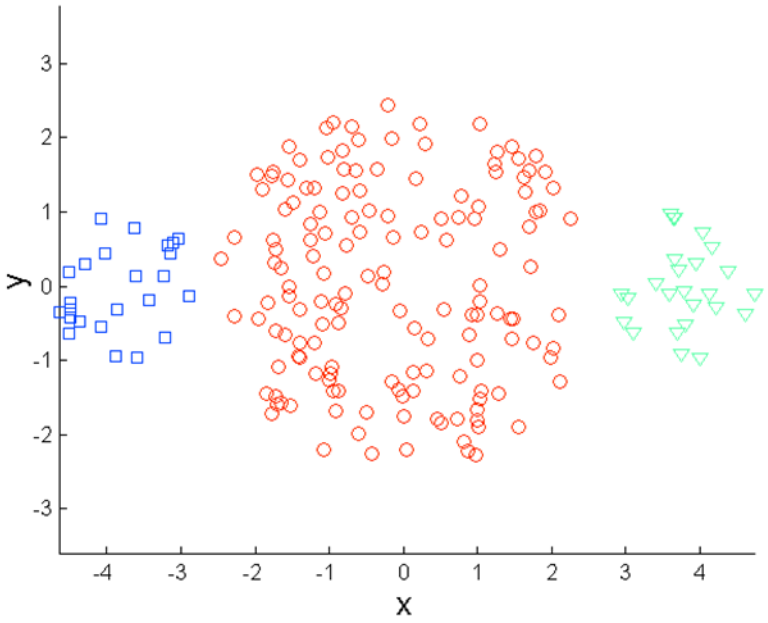


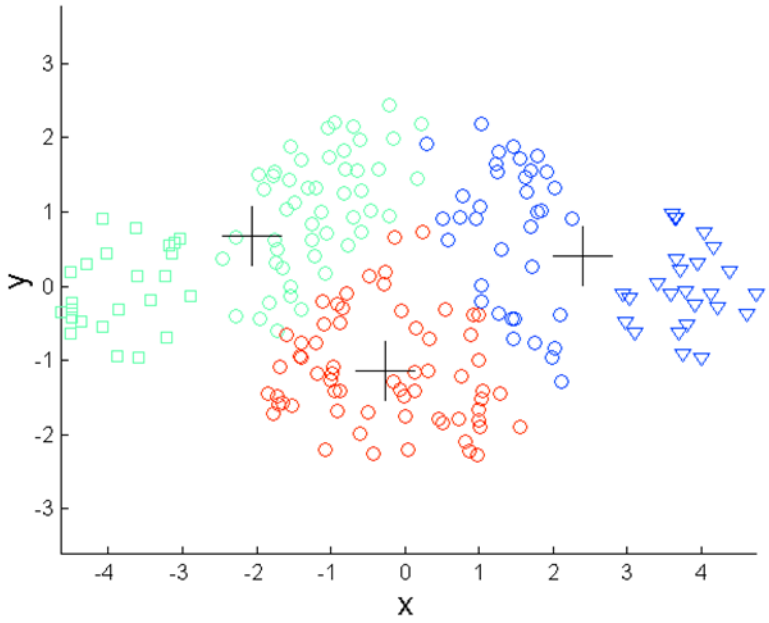
Pilsung Kang and Sungzoon Cho. (2009). K-Means clustering seeds initialization based on centrality, sparsity, and isotropy. *The 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2009)*, Burgos, Spain. E. Corchado and H. Yin (Eds.), ***Lecture Notes in Computer Science LNCS 5788***, 109-117.

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

- 문제점 1: 서로 다른 크기의 군집을 잘 찾아내지 못함

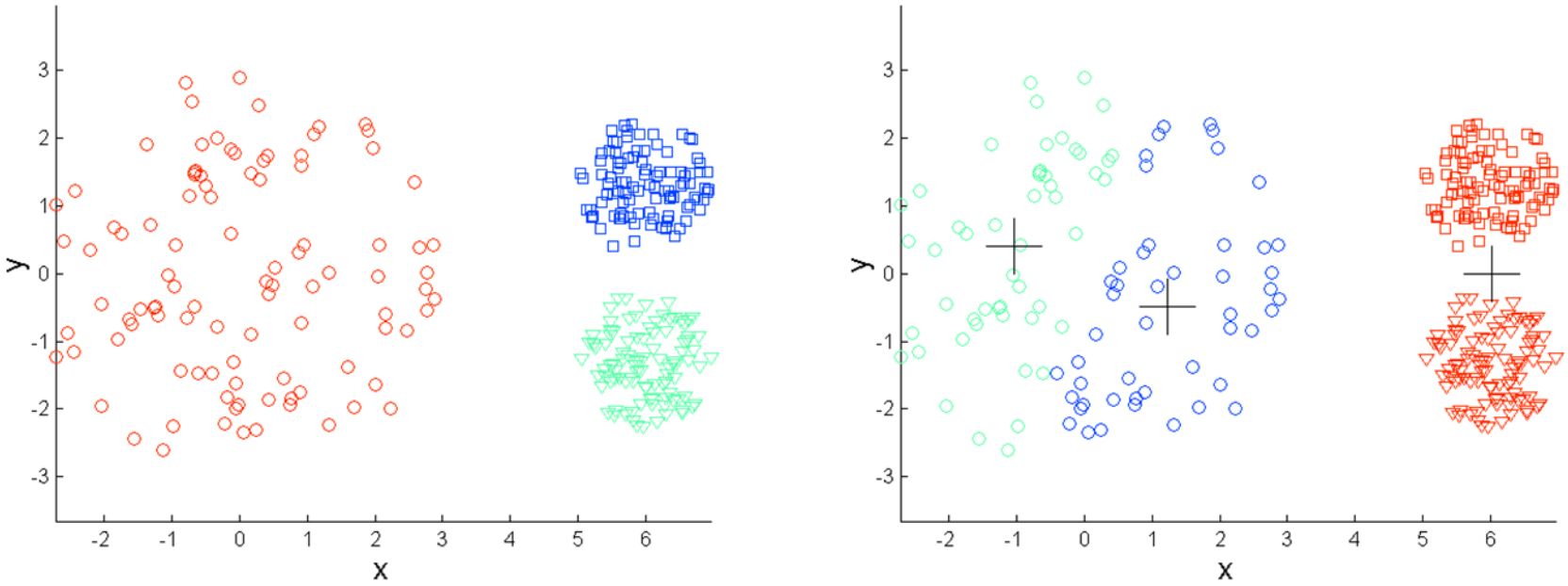




K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

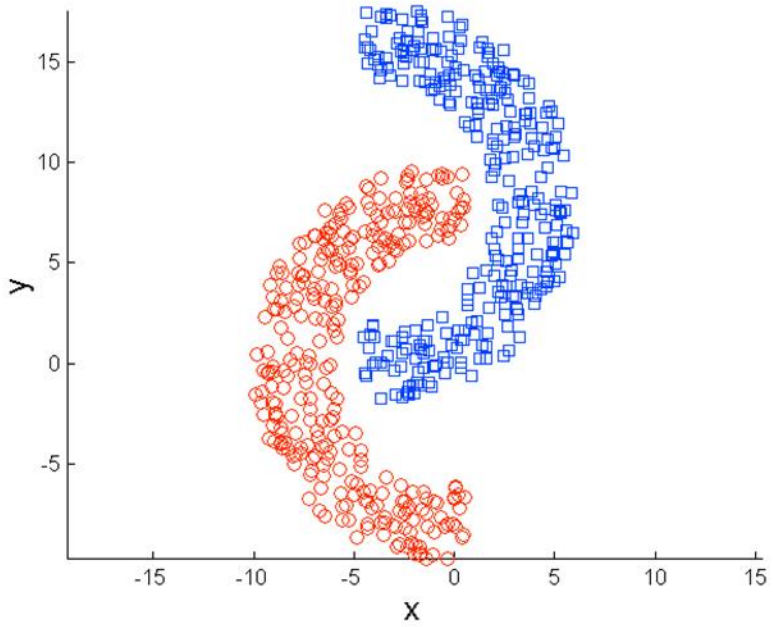
- 문제점 2: 서로 다른 밀도의 군집을 잘 찾아내지 못함

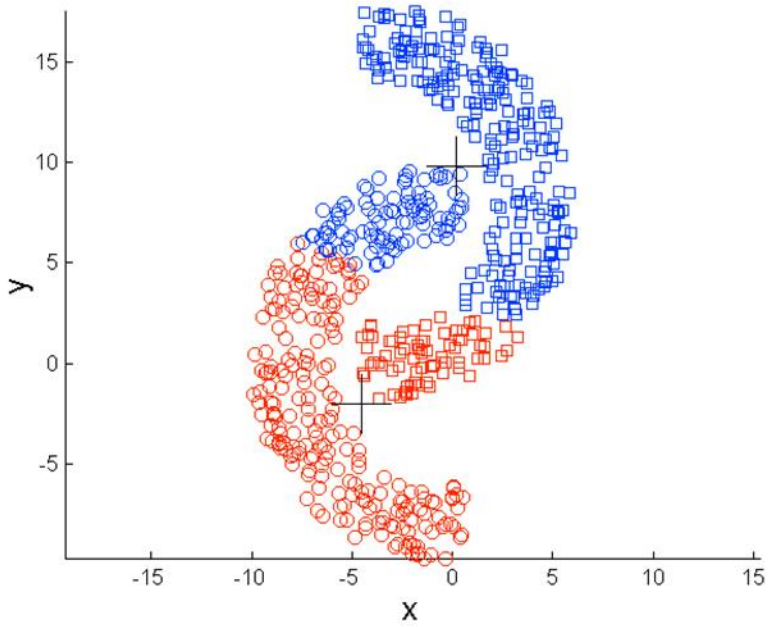


K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

- 문제점 3: 구형이 아닌 형태의 군집을 판별하기 어려움



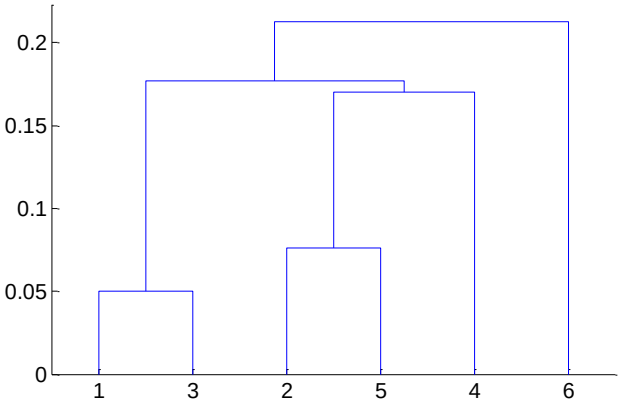


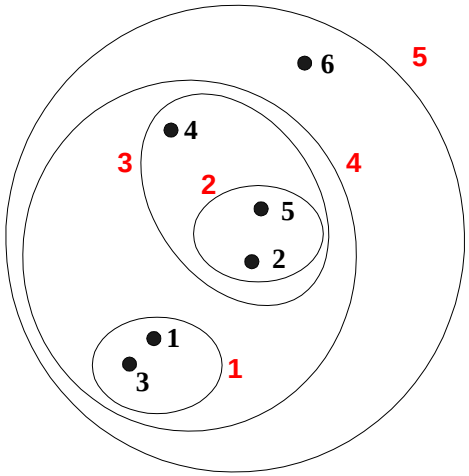
목차

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화

- 계층적 트리모형을 이용하여 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합
- 덴드로그램(Dentrogram)을 통해 시각화 가능
  - ✓Dendrogram: 개체/군집들이 결합되는 순서를 나타내는 트리형태의 구조





계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화의 장점

- 사전에 군집의 수를 정하지 않아도 수행 가능
  - ✓Dendrogram이 생성된 후 적절한 수준에서 자르면 그에 해당하는 군집화 결과 생성
- 특정 분야(domain)에서는 이 dendrogram이 유의미한 계통체계(taxonomies)를 표현하기도 함

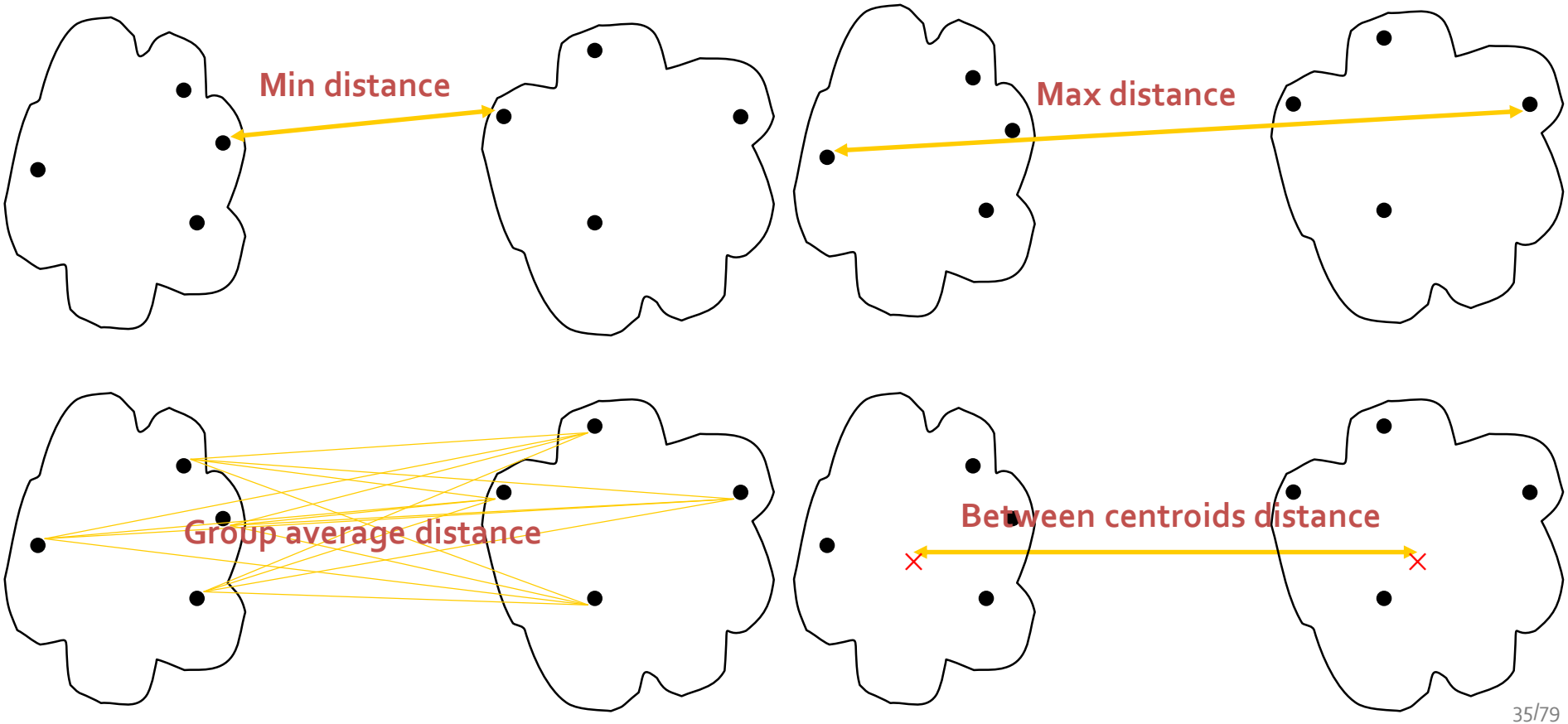
❖ 계층적 군집화의 두 가지 방식

- 상향식 군집화: Agglomerative clustering
  - ✓초기에 모든 개체들을 개별적인 군집으로 가정
  - ✓각 단계에서 유사한 개체/군집 결합 → 모든 개체들이 하나의 군집으로 통합되면 완료
- 하향식 군집화: Divisive clustering
  - ✓모든 개체가 하나의 군집으로 이루어진 상태에서 출발
  - ✓각 단계에서 가장 유의미하게 구분되는 지점을 판별하여 지속적으로 데이터를 분할

계층적 군집화: Hierarchical Clustering

❖ 상향식 군집화 알고리즘

- 핵심 수행 절차: 두 군집 사이의 유사도/거리 측정
  - ✓ Min, max, group average, between centroid, etc.

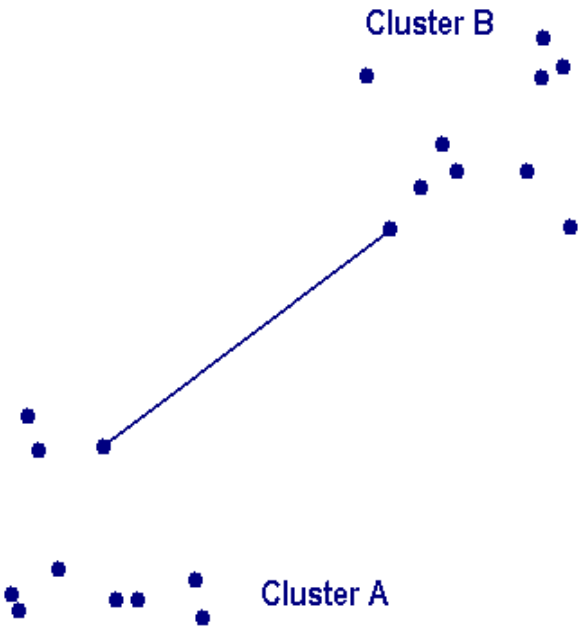


계층적 군집화: Hierarchical Clustering

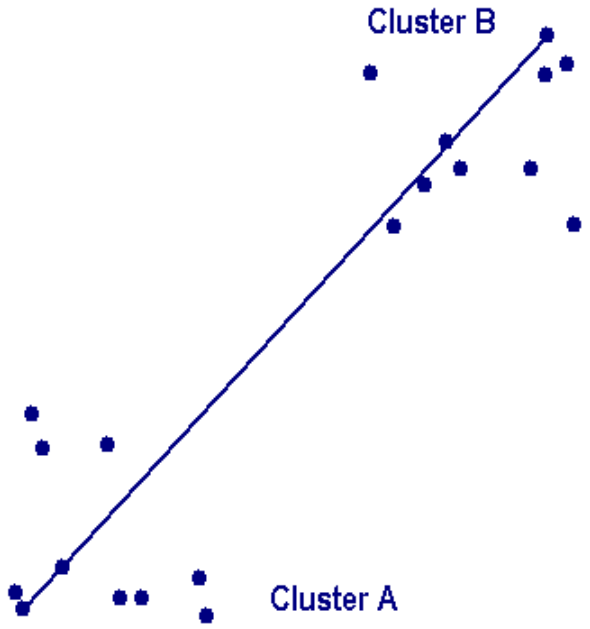
❖ 상향식 군집화 알고리즘

- Single linkage (minimum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 가까운 값을 군집간 거리로 정의
- Complete linkage (maximum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 먼 값을 군집간 거리로 정의

Single linkage



Complete linkage

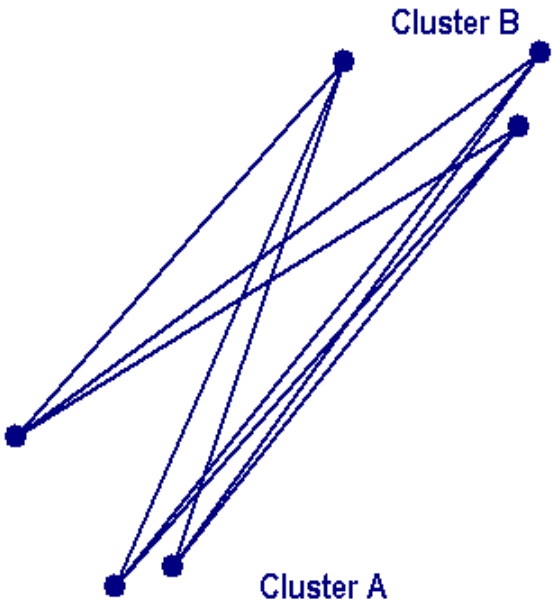


계층적 군집화: Hierarchical Clustering

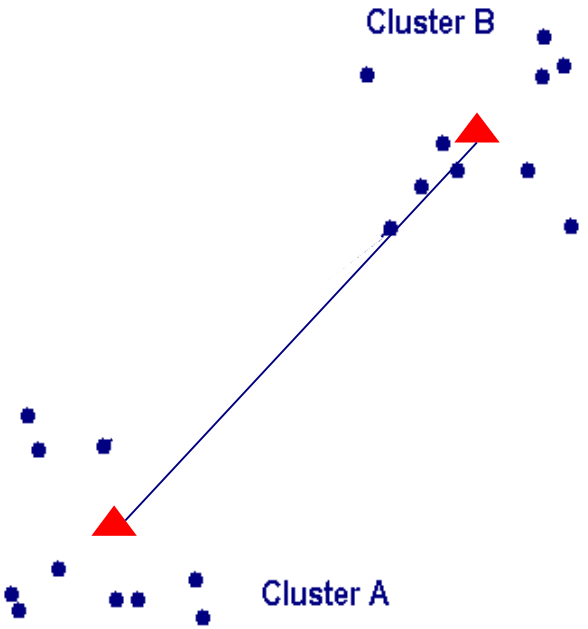
❖ 상향식 군집화 알고리즘

- Average linkage (mean distance): 각 군집에 속한 개체들 사이의 거리 평균값을 군집간 거리로 정의
- Centroid linkage (distance between centroids): 각 군집의 중심간 거리를 군집간 거리로 정의

Average linkage



Centroid linkage



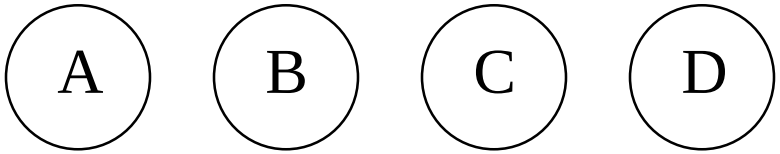
계층적 군집화: Hierarchical Clustering

- 상향식 군집화 알고리즘
  - 1단계: 모든 개체를 개별 군집으로 정의하고 군집간 거리 행렬 계산
  - 2단계: 다음 절차를 반복
    - ✓2-1단계: 가장 가까운 두 개의 군집을 하나의 군집으로 통합
    - ✓2-2단계: 군집간 거리 행렬 업데이트
  - 종료 조건: 모든 개체가 하나의 군집으로 통합되면 종료

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Initial Data Items



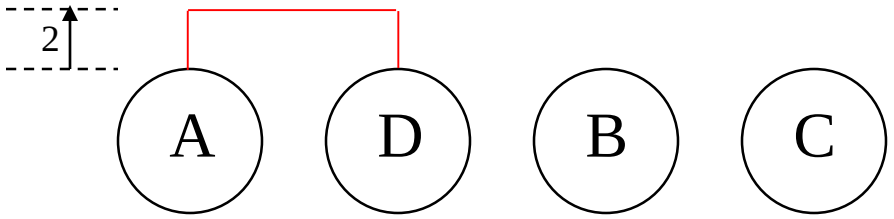
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



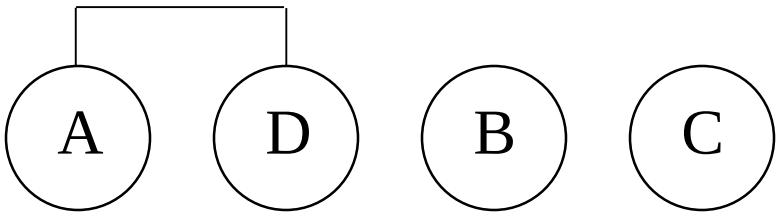
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



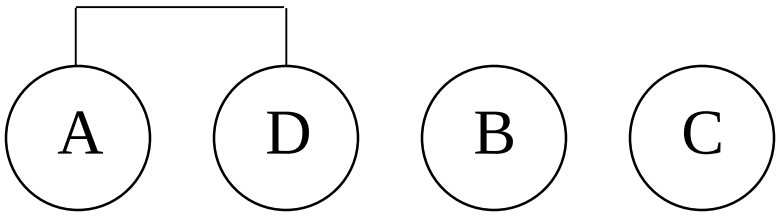
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



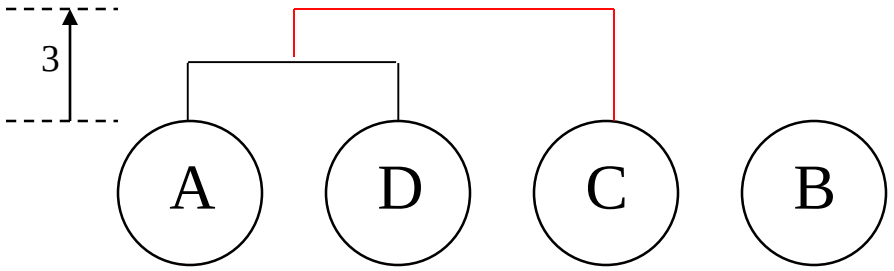
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



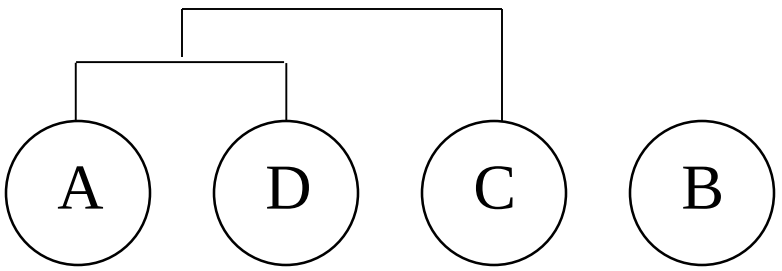
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



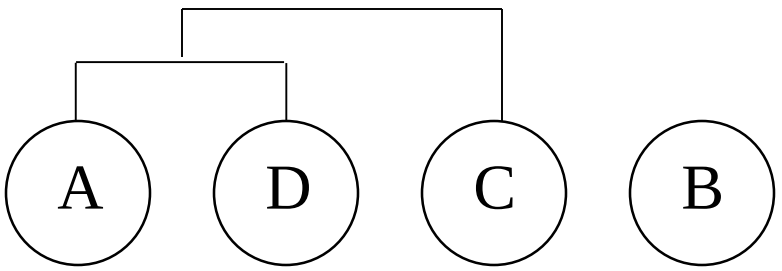
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



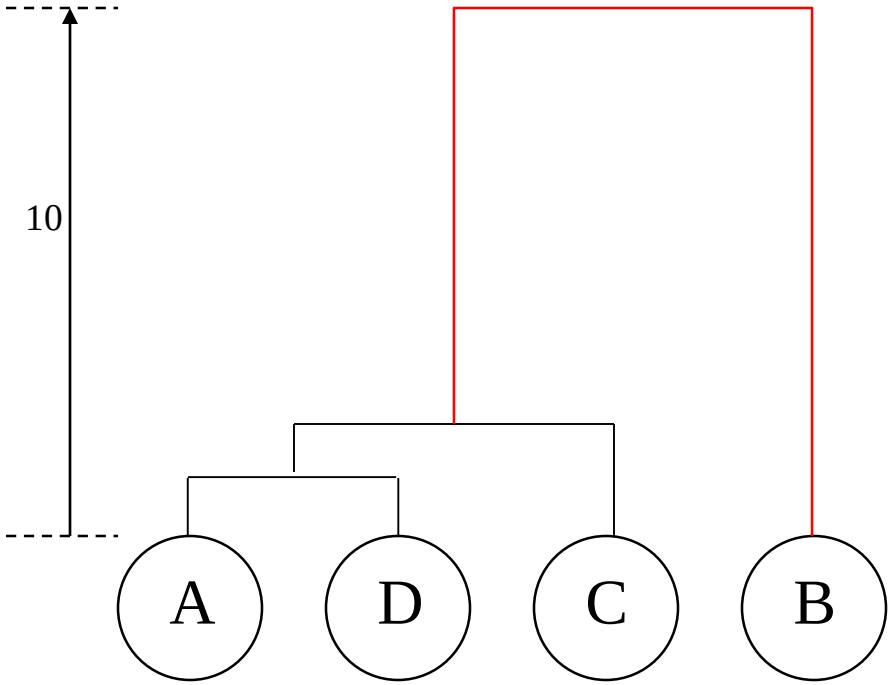
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



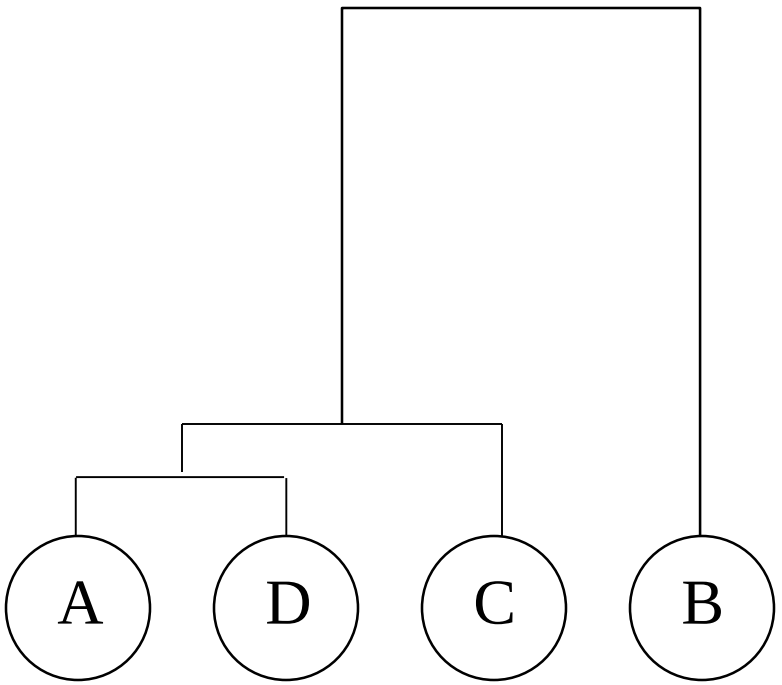
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Final Result

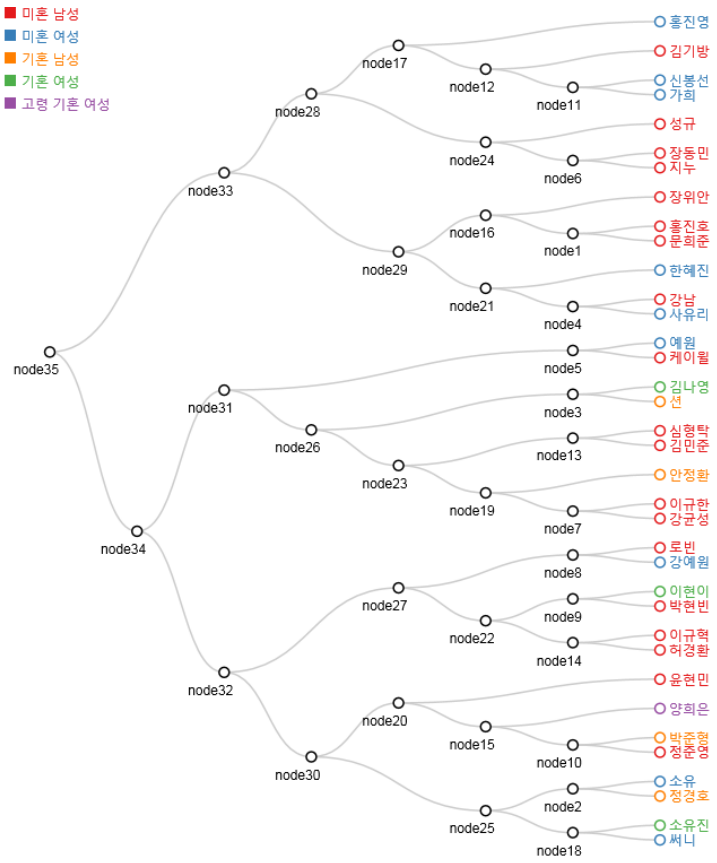


Distance Matrix

Dist	AD CB			
AD CB				

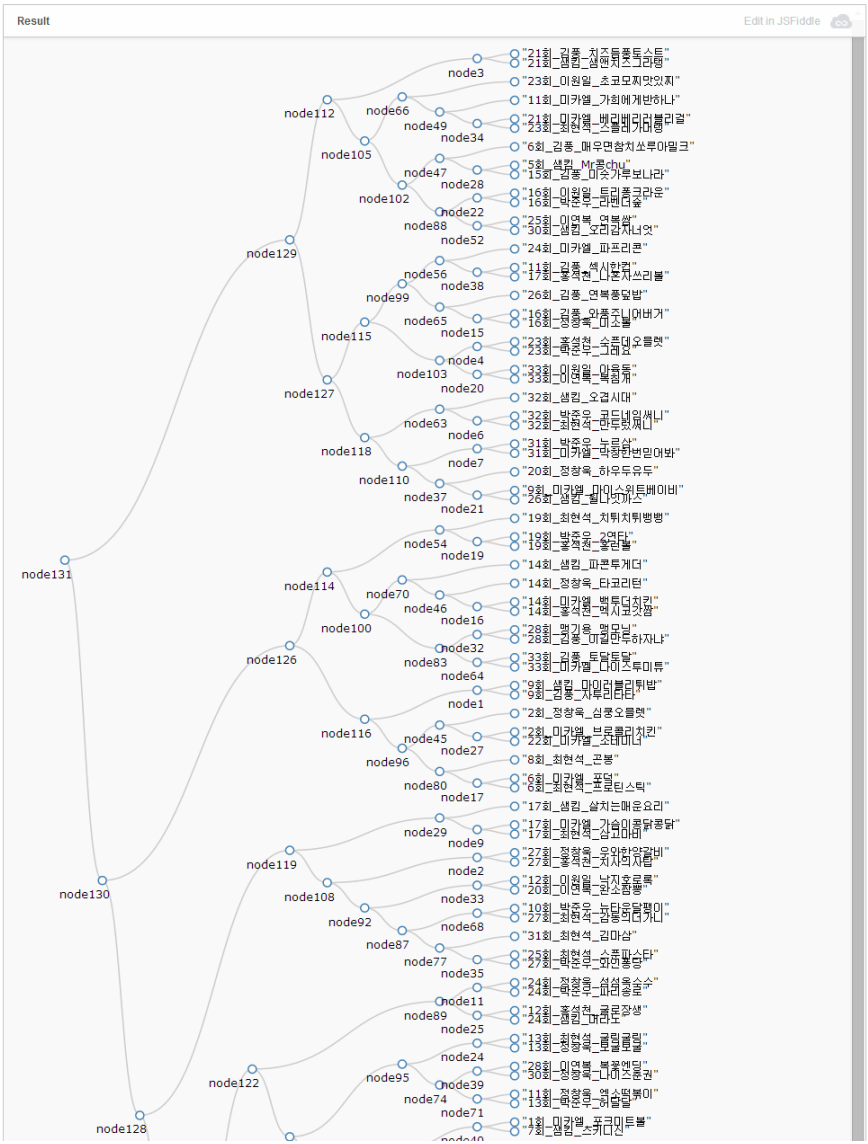
계층적 군집화: Hierarchical Clustering

❖ 냉장고를 부탁해



냉장고 재료를 이용한 게스트 군집화

레시피 계층적 군집화 분석



목차

R 실습: K-Means Clustering

- Dataset: Wine dataset from UCI machine learning repository
  - 이탈리아의 특정 지역에서 생산된 세 품종의 포도를 이용하여 생산한 178종의 와인이 함유하는 구성 성분에 대한 데이터

UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Wine Data Set

Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	733901

Source:

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,
16147 Genoa, Italy.

Donor:

Stefan Aeberhard, email: stefan '@' coral.cs.jcu.edu.au

R 실습: K-Means Clustering

- Dataset: Wine dataset from UCI machine learning repository
  - 종속변수(1열): 와인을 생산한 포도의 품종
  - 설명변수(2-14열)
    - ✓ 1) Alcohol
    - ✓ 2) Malic acid
    - ✓ 3) Ash
    - ✓ 4) Alcalinity of ash
    - ✓ 5) Magnesium
    - ✓ 6) Total phenols
    - ✓ 7) Flavanoids
    - ✓ 8) Nonflavanoid phenols
    - ✓ 9) Proanthocyanins
    - ✓ 10)Color intensity
    - ✓ 11)Hue
    - ✓ 12)OD280/OD315 of diluted wines
    - ✓ 13)Proline

R 실습: K-Means Clustering

❖ 필요 패키지 설치 및 불러오기

```
# Package for cluster validity
install.packages("clValid")
install.packages("plotrix")
library(clValid)
library(plotrix)
```

- "clValid" 패키지
  - ✓여러 가지 군집타당성 지표들을 계산할 수 있는 기능을 포함
- "plotrix" 패키지
  - ✓Radar chart를 도시할 수 있는 패키지

❖ K-평균 군집화는 다양한 패키지에서 제공

- stats, kml, kml3d, RSKC, skmeans, sparcl, etc.
- R에서 기본 제공되는 기능 사용

R 실습: K-Means Clustering

❖ 데이터 불러오기 및 전처리

```
# Part 1: K-Means Clustering ----------------------------------------------
# Load the Wine dataset
wine <- read.csv("wine.csv")

# Remove the class label
wine_class <- wine[,1]
wine_x <- wine[,-1]

# data scaling
wine_x_scaled <- scale(wine_x, center = TRUE, scale = TRUE)
```

- 데이터 전처리
  - ✓첫 번째 열은 종속변수에 해당하므로 군집화에는 사용하지 않음
  - ✓KMC는 개체 사이의 거리를 계산하는 과정이 포함되어 있으므로 데이터 정규화는 필수적으로 수행되어야 함

R 실습: K-Means Clustering

❖ 군집 타당성 지표를 통한 최적의 군집 수 결정

```
# Evaluating the cluster validity measures
wine_clValid <- clValid(wine_x_scaled, 2:10, clMethods = "kmeans",
                        validation = c("internal", "stability"))
summary(wine_clValid)
```

- clValid( ): 여러 가지 군집 타당성 지표를 계산해주는 함수
  - ✓첫 번째 인자: 군집화에 사용할 데이터셋
  - ✓두 번째 인자: 후보 군집의 수(본 실험에서는 2부터 10까지 총 9가지 경우 탐색)
  - ✓세 번째 인자: 군집화 알고리즘(KMC 이외의 다양한 군집화 알고리즘 사용 가능)
  - ✓네 번째 인자: 산출할 타당성 지표의 카테고리 (본 실험에서는 internal, stability 두 카테고리에 속하는 타당성 지표들에 대한 값을 산출)

R 실습: K-Means Clustering

❖ 군집 타당성 지표를 통한 최적의 군집 수 결정

- Dunn index와 Silhouette 두 가지 지표 모두 군집의 수는 3개가 최적으로 판단

```
> summary(wine_clValid)

Clustering Methods:
 kmeans

Cluster sizes:
 2 3 4 5 6 7 8 9 10

Validation Measures:
                            2        3        4        5        6        7        8        9       10

kmeans APN             0.1255   0.0470   0.1851   0.1392   0.1178   0.1634   0.2325   0.2543   0.2304
       AD              4.2577   3.6137   3.6186   3.4572   3.3486   3.3064   3.2889   3.2236   3.1174
       ADM             0.6454   0.2231   0.7547   0.5135   0.4513   0.5565   0.7912   0.8512   0.7698
       FOM             0.9139   0.7842   0.7728   0.7548   0.7435   0.7424   0.7264   0.7329   0.7157
       Connectivity   37.6512  28.0504  61.1659  76.2976  84.5433  88.4012  95.2159 109.1425 111.7302
       Dunn            0.1357   0.2323   0.1621   0.1900   0.2021   0.2111   0.2081   0.2081   0.2307
       Silhouette      0.2593   0.2849   0.2127   0.2656   0.2446   0.2323   0.2307   0.2210   0.2209

Optimal Scores:

             Score   Method Clusters
APN           0.0470 kmeans 3
AD            3.1174 kmeans 10
ADM           0.2231 kmeans 3
FOM           0.7157 kmeans 10
Connectivity 28.0504 kmeans 3
Dunn          0.2323 kmeans 3
Silhouette    0.2849 kmeans 3
```

R 실습: K-Means Clustering

❖ 최적의 군집 수를 사용한 KMC 수행

```
# Perform K-Means Clustering with the best K determined by Silhouette
wine_kmc <- kmeans(wine_x_scaled,3)

str(wine_kmc)
wine_kmc$centers
wine_kmc$size
wine_kmc$cluster
```

- kmeans( ): K-평균 군집화 수행 함수
  - ✓첫 번째 인자: 사용할 데이터셋
  - ✓두 번째 인자: 군집의 수
- kmeans( )가 실행되면 저장되는 대표적인 결과물은 다음과 같음
  - ✓$centers: 각 군집 중심의 좌표
  - ✓$size: 각 군집에 할당된 개체 수
  - ✓$cluster: 각 개체가 할당된 군집 번호 (주의: 군집 번호는 구분 역할만을 수행하며 다른 의미는 없음!)

R 실습: K-Means Clustering

❖ 실제 범주와 할당된 군집 비교

```
# Compare the cluster info. and class labels
real_class <- wine_class
kmc_cluster <- wine_kmc$cluster
table(real_class, kmc_cluster)
```

```
> table(real_class, kmc_cluster)
          kmc_cluster
real_class  1  2  3
         1  0  0 59
         2 65  3  3
         3  0 48  0
```

- 범주 정보를 사용하지 않은 군집화지만
  - ✓최적의 군집 수는 실제 범주의 수와 같은 3으로 판별이 되었으며,
  - ✓대부분의 군집은 하나의 범주에 속하는 개체들이 대다수를 이룸

R 실습: K-Means Clustering

❖ 실제 범주와 할당된 군집 비교

```
# Compare each cluster for KMC
cluster_kmc <- data.frame(wine_x_scaled, clusterID = as.factor(wine_kmc$cluster))
kmc_summary <- data.frame()

for (i in 1:(ncol(cluster_kmc)-1)){
        kmc_summary = rbind(kmc_summary, tapply(cluster_kmc[,i],
                            cluster_kmc$clusterID, mean))
}
colnames(kmc_summary) <- paste("cluster", c(1:3))
rownames(kmc_summary) <- colnames(wine_x)
kmc_summary
```

- 정규화된 데이터와 각 개체가 할당된 군집 정보로 구성된 데이터프레임 생성
- kmc_summary 데이터 프레임
  - ✓각 군집들에 대해 각 변수값의 평균을 저장
  - ✓열 이름은 cluster1, cluster2, cluster3을 할당
  - ✓행 이름은 wine_x데이터셋의 열 이름(원래 설명변수 이름)을 할당

R 실습: K-Means Clustering

❖ 군집별 속성 비교

```
> kmc_summary
                                cluster 1   cluster 2  cluster 3
Alcohol.2                     -0.92346686  0.16444359  0.8328826
Malic.acid.                   -0.39293312  0.86909545 -0.3029551
Ash.                          -0.49312571  0.18637259  0.3636801
Alcalinity.of.ash.             0.17012195  0.52289244 -0.6084749
Magnesium.                    -0.49032869 -0.07526047  0.5759621
Total.phenols.                -0.07576891 -0.97657548  0.8827472
Flavanoids.                    0.02075402 -1.21182921  0.9750690
Nonflavanoid.phenols          -0.03343924  0.72402116 -0.5605085
Proanthocyanins.               0.05810161 -0.77751312  0.5786543
Color.intensity.              -0.89937699  0.93889024  0.1705823
Hue                            0.46050459 -1.16151216  0.4726504
OD280.OD315.of.diluted.wines.  0.27000254 -1.28877614  0.7770551
Proline.                      -0.75172566 -0.40594284  1.1220202
```

- 알코올 도수는 군집 3 > 군집 2> 군집 1순
- Malic acid는 군집 2가 평균적으로 높으며 군집 1과 3은 큰 차이를 나타내지 않음
- Flavornoids는 군집 3 > 군집 1 > 군집 2 순이며 상대적으로 그 차이가 두드러짐

R 실습: K-Means Clustering

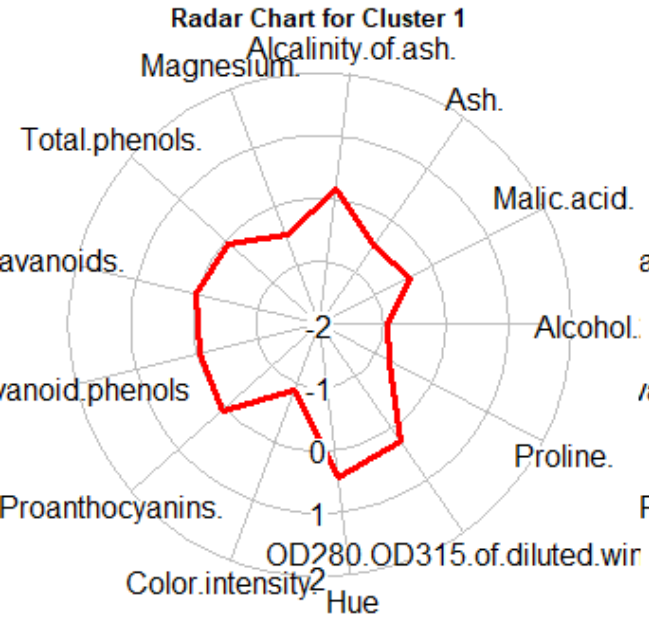
❖ 각 군집별 변수값의 평균을 Radar Chart로 도시

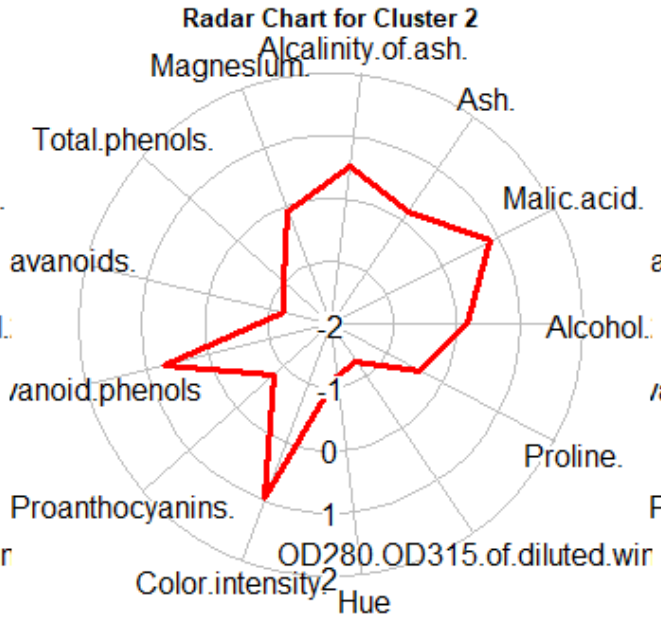
```
# Radar chart
par(mfrow = c(1,3))
for (i in 1:3){
          plot_title <- paste("Radar Chart for Cluster", i, sep=" ")
          radial.plot(kmc_summary[,i], labels = rownames(kmc_summary),
                              radial.lim=c(-2,2), rp.type = "p", main = plot_title,
                              line.col = "red", lwd = 3, show.grid.labels=1)
}
dev.off()
```

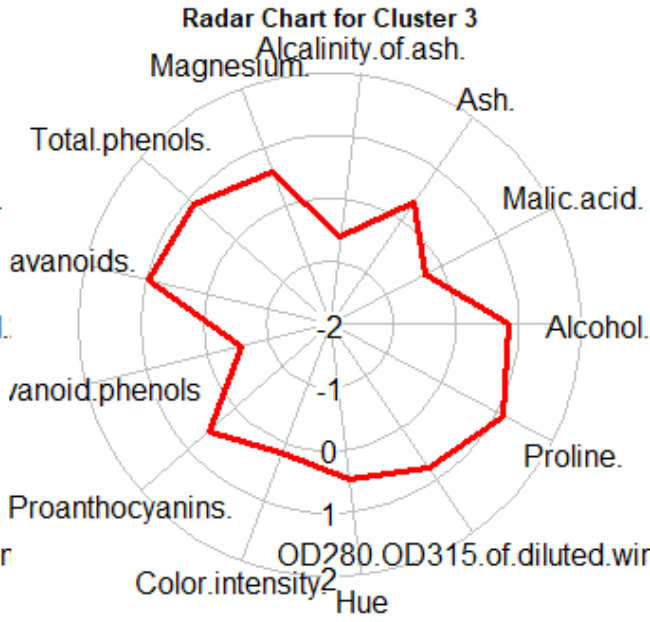
- par(mfrow = c(1,3)): 1행 3열로 구성된 그림 도시를 선언
- radial.plot( ): Radar chart를 도시해주는 함수
- dev.off( ): 앞서 정의된 par( )의 설정을 초기화하는 함수

R 실습: K-Means Clustering

- 각 군집별 변수값의 평균을 Radar Chart로 도시
  - 세 군집에 대해 13개의 변수들의 평균값을 도시한 결과







R 실습: K-Means Clustering

- 군집 1과 군집 2에 대한 통계적 가설 검정 수행
  - H0: 군집 1의 변수 평균 = 군집 2의 변수 평균
  - H1:
    - ✓첫 번째 열: 군집 1의 변수 평균 ≠ 군집 2의 변수 평균
    - ✓두 번째 열: 군집 1의 변수 평균 > 군집 2의 변수 평균
    - ✓세 번째 열: 군집 1의 변수 평균 < 군집 2의 변수 평균

R 실습: K-Means Clustering

❖ 군집 1과 군집 2에 대한 통계적 가설 검정 수행

```
# Compare the first and the second cluster
kmc_cluster1 <- wine_x[wine_kmc$cluster == 1,]
kmc_cluster2 <- wine_x[wine_kmc$cluster == 2,]

# t_test_result
kmc_t_result <- data.frame()
for (i in 1:13){
        kmc_t_result[i,1] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                    alternative = "two.sided")$p.value
        kmc_t_result[i,2] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                    alternative = "greater")$p.value
        kmc_t_result[i,3] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                    alternative = "less")$p.value
}
kmc_t_result
```

- t-test( ): 두 집단에 대한 Student's t-test를 수행해주는 함수
  - ✓첫 번째 인자: 집단 1의 변수 값, 두 번째 인자: 집단 2의 변수 값
  - ✓세 번째 인자: 가설("two-sided": 양측 검정, "greater" & "less": 단측 검정)
  - ✓$p.value: 가설검정 결과의 유의확률

R 실습: K-Means Clustering

❖ 군집 1과 군집 2에 대한 통계적 가설 검정 수행

```
> kmc_t_result
                 V1           V2           V3
1   1.018635e-14 1.000000e+00 5.093173e-15
2   1.475032e-10 1.000000e+00 7.375160e-11
3   1.075500e-04 9.999462e-01 5.377501e-05
4   2.114410e-02 9.894280e-01 1.057205e-02
5   1.118367e-02 9.944082e-01 5.591835e-03
6   3.791205e-10 1.895603e-10 1.000000e+00
7   2.146292e-25 1.073146e-25 1.000000e+00
8   7.509477e-05 9.999625e-01 3.754739e-05
9   7.432607e-07 3.716303e-07 9.999996e-01
10 4.041358e-18 1.000000e+00 2.020679e-18
11 3.690316e-22 1.845158e-22 1.000000e+00
12 1.850143e-30 9.250715e-31 1.000000e+00
13 2.031718e-05 9.999898e-01 1.015859e-05
```

유의수준 0.01에서 군집 1의 해당 변수 값의 평균이 군집 2의 변수값 평균보다 큼

유의수준 0.01에서 군집 1의 해당 변수 값의 평균이 군집 2의 변수값 평균보다 작음

R 실습: 계층적 군집화

❖ 데이터 불러오기 및 전처리

```
# Part 2: Hierarchical Clustering ---------------------------------------
ploan <- read.csv("Personal Loan.csv")
ploan_x <- ploan[,-c(1,5,10)]
ploan_x_scaled <- scale(ploan_x, center = TRUE, scale = TRUE)
```

- 개인신용대출 데이터셋 사용
  - ✓ 1열(ID관련 변수)과 5열(zip code 변수) 제거
  - ✓ 10열은 원래 종속변수이므로 군집화에 사용하지 않음
  - ✓ 개체간 유사도/거리를 계산하는 과정이 포함되어 있으므로 반드시 정규화 수행

R 실습: 계층적 군집화

❖ 거리행렬 생성

```
# Compute the similarity using the spearman coefficient
cor_Mat <- cor(t(ploan_x_scaled), method = "spearman")
dist_ploan <- as.dist(1-cor_Mat)
```

- 두 개체간 유사성을 계산하기 위해서 상관계수를 이용
  - ✓ t( ) 함수는 원래 데이터에 transpose(행과 열을 바꾸는 작업)를 수행
  - ✓ cor( ) 함수는 상관계수를 구하는 함수이며 두 번째 인자인 method에 "spearman"옵션을 사용하여 이상치에 덜 민감한 스피어만 순위상관계수를 사용
- as.dist( ): 효율적인 거리행렬 정보 저장을 지원하는 함수
  - ✓ 상관계수는 유사성 지표이고 계층적 군집화는 거리정보를 이용하여 수행되기 때문에 1에서 상관계수를 뺀 값을 거리정보로 취급
  - ✓ as.dist( )는 중복된 거리 정보를 제거하여 upper 또는 lower triangle 정보만을 저장

R 실습: 계층적 군집화

❖ 계층적 군집화 수행

```
# Perform hierarchical clustering
hr <- hclust(dist_ploan, method = "complete", members=NULL)
```

- hclust( ): 계층적 군집화 수행 함수
  - ✓첫 번째 인자: 거리 행렬
  - ✓두 번째 인자: 군집간 거리 계산 옵션(본 실험에서는 maximum distance인 "complete" linkage 사용
  - ✓계층적 군집화는 군집의 수를 처음에 결정할 필요가 없음

R 실습: 계층적 군집화

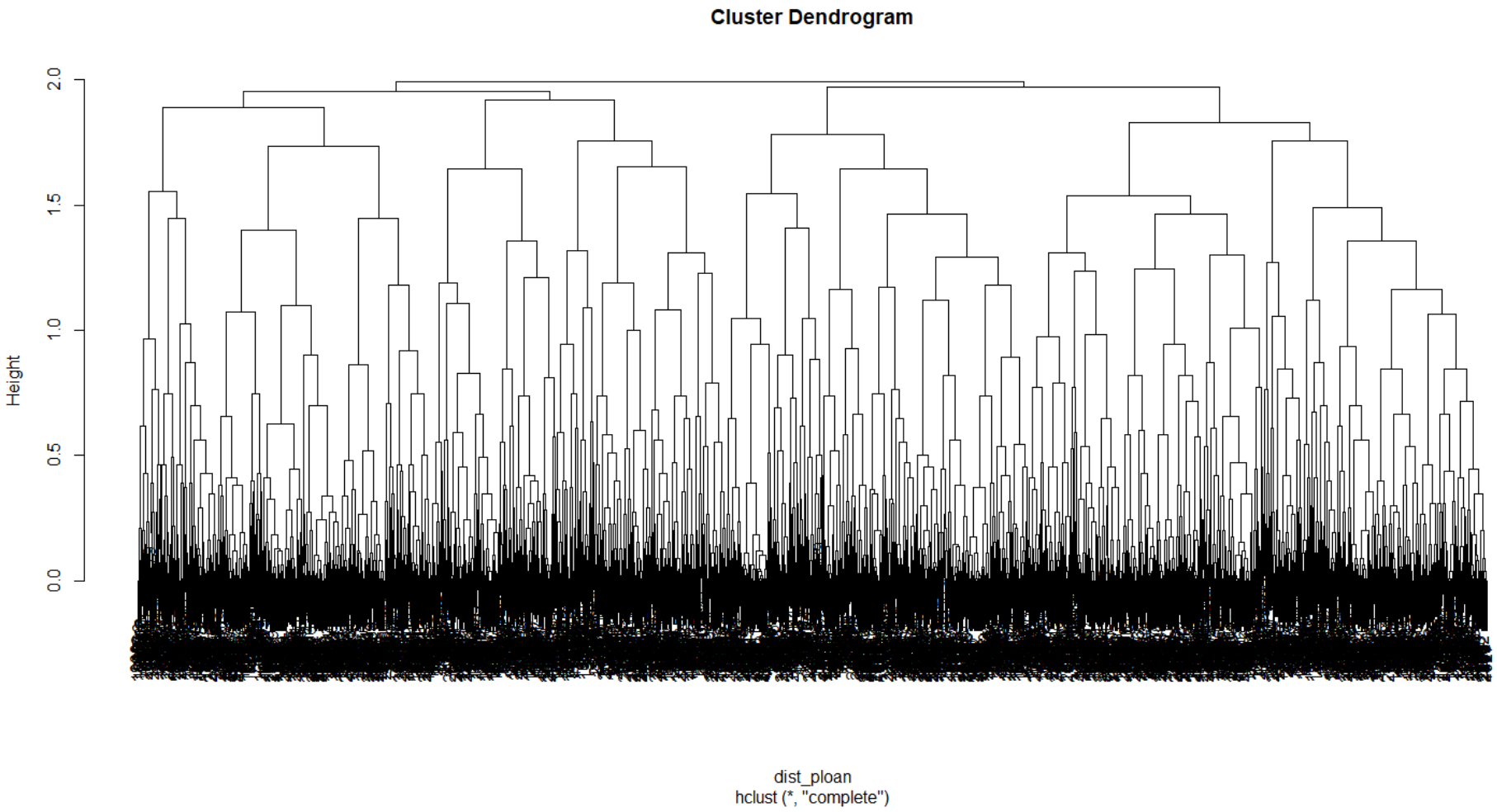
❖ 덴드로그램 도시

```
# plot the results
plot(hr)
plot(hr, hang = -1)
plot(as.dendrogram(hr), edgePar=list(col=3, lwd=4), horiz=T)
```

- 덴드로그램을 도시하는 다양한 옵션 존재
  - ✓ 그림 1: 가장 기본적인 덴드로그램 (다음 페이지 참고)
  - ✓ 그림 2: 말단 개체들을 병합된 순서에 따라 높이를 다르게 표시
  - ✓ 그림 3: 그림 1과 2는 상하 방식의 덴드로그램인 반면 그림 3은 좌우 방식의 덴드로그램 도시, 선의 굵기와 색상도 변경 가능

R 실습: 계층적 군집화

❖ 덴드로그램 도시



R 실습: 계층적 군집화

❖ 군집 생성

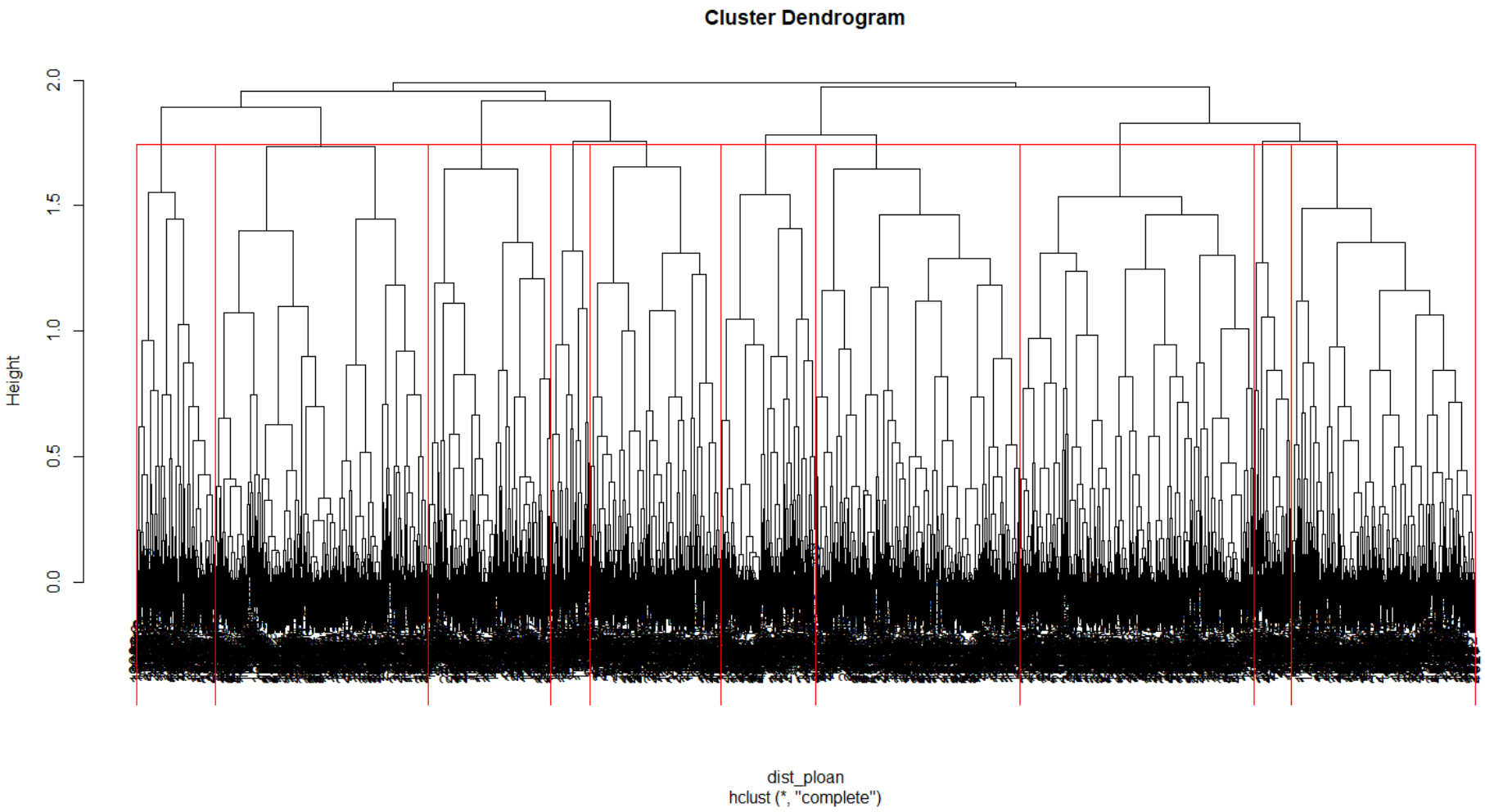
```
# Find the clusters
mycl <- cutree(hr, k=10)
mycl

plot(hr)
rect.hclust(hr, k=10, border="red")
```

- cutree( ): 군집을 생성하는 함수
  - ✓첫 번째 인자: 덴드로그램
  - ✓두 번째 인자: k=10 → 10개의 군집을 생성
- rect.hclust( ): 덴드로그램과 함께 생성한 군집을 직사각형으로 표시하는 기능을 제공 (다음 페이지 그림 참조)

R 실습: 계층적 군집화

❖ 군집 생성



R 실습: 계층적 군집화

❖ 군집별 특징 비교

```
# Compare each cluster for HC
ploan_hc <- data.frame(ploan_x_scaled, ploanYN = ploan[,10],
                              clusterID = as.factor(mycl))
hc_summary <- data.frame()
for (i in 1:(ncol(ploan_hc)-1)){
         hc_summary = rbind(hc_summary, tapply(ploan_hc[,i],
                              ploan_hc$clusterID, mean))
}
colnames(hc_summary) <- paste("cluster", c(1:10))
rownames(hc_summary) <- c(colnames(ploan_x), "LoanRatio")
hc_summary
```

- KMC와 유사하게 각 군집별로 변수들의 평균값을 산출
- 이에 더하여 각 군집별 대출 고객 비율(LoanRatio) 계산
  - ✓Target marketing의 관점에서 볼 때, 제한된 예산을 사용할 경우 LoanRatio가 높은 군집부터 캠페인 실시
  - ✓LoanRatio가 가장 높은 군집과 가장 낮은 군집에 대한 비교 분석 수행 가능

R 실습: 계층적 군집화

❖ 군집별 특징 비교

```
> hc_summary
                     cluster 1   cluster 2   cluster 3   cluster 4    cluster 5 cluster 6   cluster 7   cluster 8   cluster 9 cluster 10
Age                -0.978375630  0.62768747 -1.00065813  0.45556464  0.623411101 0.3606053  0.83631077 -0.94640912  0.23039152 -0.27857180
Experience         -0.984955242  0.63865510 -1.01149000  0.45826338  0.594577161 0.3583185  0.84354883 -0.93187619  0.22877326 -0.27319032
Income             -0.041720588  0.11353167 -0.34249916  0.67417711  0.055576655 0.2654431 -0.76794149  0.68688171 -0.30617789  0.99685044
Family              0.249749572 -0.06963678  0.64165613 -0.81485549  0.758429117 -0.2598958  0.09881633 -0.61816862 -0.84443724 -0.42263030
CCAvg               0.056049820  0.07592153 -0.27792518  0.55790241  0.002870789 0.3816039 -0.55877036  0.47138980 -0.43458985  0.01431525
Education           0.219734047 -0.33718946  0.21399718 -0.13166151  0.168847347 0.0920721  0.02249967 -0.20870083  0.82666013 -0.77272322
Mortgage           -0.109905334 -0.04520828 -0.09913763  0.04240734  0.412406634 -0.3300013 -0.16572200  0.12002376  0.89665284  1.20219168
Securities.Account  0.002768788  0.02273880  0.04403728  0.13838377 -0.175403903 -0.3507727  0.13171186  0.13469351 -0.12216694 -0.21924611
CD.Account         -0.197847765 -0.21063348 -0.07349502 -0.04132722 -0.229615630 0.2672237  0.03286815  0.48516412 -0.08077930  0.98776848
Online             -1.200618309 -1.20906773  0.76690876  0.81151795  0.624917820 0.3859185  0.51306132  0.67151852 -1.10189473  0.42941455
CreditCard         -0.044161882 -0.17087863 -0.13667044 -0.59150572 -0.639594361 1.5628656 -0.25222704 -0.05804595  0.11553475  1.26115872
LoanRatio           0.098837209  0.10755149  0.07874016  0.17467249  0.102739726 0.1147541  0.01005025  0.21348315  0.07142857  0.20547945
```

- LoanRatio의 관점에서 보면
  - ✓군집 7이 1.01%로 가장 반응률이 낮고 군집 8이 21.35%로 가장 반응률이 높음
  - ✓반응률 차이는 약 25배

R 실습: 계층적 군집화

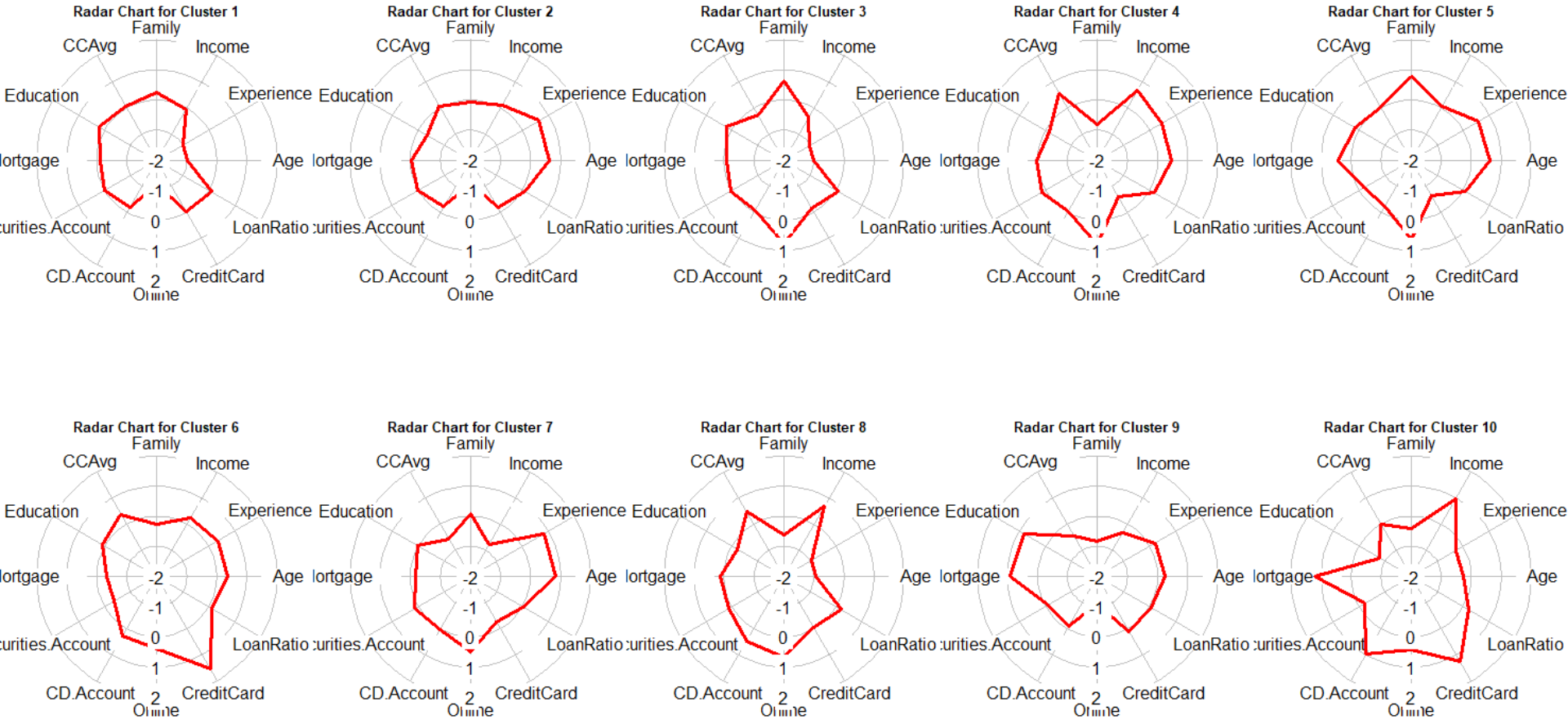
❖ 군집별 특징 비교 (Radar Chart 도시)

```
# Radar chart
par(mfrow = c(2,5))
for (i in 1:10){
          plot_title <- paste("Radar Chart for Cluster", i, sep=" ")
          radial.plot(hc_summary[,i], labels = rownames(hc_summary),
                              radial.lim=c(-2,2), rp.type = "p", main = plot_title,
                              line.col = "red", lwd = 3, show.grid.labels=1)
}
dev.off()
```

- KMC와 유사하게 각 군집별로 변수값들의 평균을 이용한 Radar Chart 도시

R 실습: 계층적 군집화

❖ 군집별 특징 비교 (Radar Chart 도시)



R 실습: 계층적 군집화

- 신용대출 이용률이 가장 낮은 군집(7)과 높은 군집(8) 비교
  - H0: 군집 7의 변수 평균 = 군집 8의 변수 평균
  - H1:
    - ✓첫 번째 열: 군집 7의 변수 평균 ≠ 군집 8의 변수 평균
    - ✓두 번째 열: 군집 7의 변수 평균 > 군집 8의 변수 평균
    - ✓세 번째 열: 군집 7의 변수 평균 < 군집 8의 변수 평균

R 실습: 계층적 군집화

- 신용대출 이용률이 가장 낮은 군집(7)과 높은 군집(8) 비교

```
# Compare the cluster 7 & 8
hc_cluster7 <- ploan_hc[ploan_hc$clusterID == 7, c(1:11)]
hc_cluster8 <- ploan_hc[ploan_hc$clusterID == 8, c(1:11)]

# t_test_result
hc_t_result <- data.frame()
for (i in 1:11){
        hc_t_result[i,1] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                            alternative = "two.sided")$p.value
        hc_t_result[i,2] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                            alternative = "greater")$p.value
        hc_t_result[i,3] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                            alternative = "less")$p.value
}
hc_t_result
```

R 실습: 계층적 군집화

- 신용대출 이용률이 가장 낮은 군집(7)과 높은 군집(8) 비교
  - 군집 7에서 더 높은 값을 갖는 변수: Age, Experience, Family, Educaton
  - 군집 8에서 더 높은 값을 갖는 변수: Income, CCAvg, Mortgage, CD.Account, Online, Creditcard

```
> hc_t_result
                v1           v2           v3
1   1.216994e-95 6.084971e-96 1.000000e+00
2   5.671113e-96 2.835557e-96 1.000000e+00
3   2.573588e-41 1.000000e+00 1.286794e-41
4   5.199496e-21 2.599748e-21 1.000000e+00
5   2.054088e-26 1.000000e+00 1.027044e-26
6   1.147233e-02 5.736163e-03 9.942638e-01
7   5.157287e-03 9.974214e-01 2.578643e-03
8   9.770747e-01 5.114626e-01 4.885374e-01
9   6.261203e-04 9.996869e-01 3.130601e-04
10  3.591460e-03 9.982043e-01 1.795730e-03
11  2.184769e-02 9.890762e-01 1.092384e-02
```

Q & A